

# **Universidad Carlos III de Madrid**



## **Ingeniería Técnica en Informática de Gestión**

Proyecto Fin de Carrera

Desarrollo y validación de un sistema  
dinámico de adaptación de páginas web  
para acceso de terminales móviles

Autor: Javier López Prudencio

Tutor: Dr. José Ignacio Moreno Novella



## Agradecimientos

En primer lugar me gustaría dar las gracias a mi familia por el apoyo incondicional que me han prestado siempre.

Igualmente, tengo que agradecer a toda la gente con la que he compartido mi estancia en Berlín durante los 6 meses de prácticas en los que he desarrollado este proyecto. Especialmente, me gustaría agradecer a Steffen, mi tutor en la compañía, con el que he compartido despacho durante todo este tiempo, por sus consejos y por todo lo que he aprendido junto a él.

También tengo que agradecer a José Ignacio, responsable de las prácticas y tutor del proyecto, su ayuda e interés mostrado antes, durante y después de mi estancia en Alemania.

# Tabla de Contenidos

<b>CAPÍTULO 1 - MOTIVACIÓN Y OBJETIVOS .....</b>	<b>1</b>
1.1    MOTIVACIÓN.....	1
1.2    OBJETIVOS .....	3
<b>CAPÍTULO 2 - ESTADO DEL ARTE.....</b>	<b>5</b>
2.1    VISIÓN GENERAL DE LAS SOLUCIONES DE ADAPTACIÓN EXISTENTES.....	6
2.1.1    Panorámica inicial .....	6
2.1.2    Transcodificadores generales .....	7
2.1.2.1    Google Mobilizer .....	8
2.1.2.2    Mowser.....	9
2.1.2.3    Skweezer.....	11
2.1.3    Resultados generales de los transcodificadores .....	14
2.1.4    Otras formas de adaptación automática .....	15
<b>CAPÍTULO 3 - ANÁLISIS, DISEÑO Y DESARROLLO. ....</b>	<b>17</b>
3.1    PASOS PREVIOS NECESARIOS .....	17
3.1.1    Copia local del punto de acceso .....	17
3.1.2    Preparación de un servidor web que almacene la copia de la página.....	19
3.2    PRUEBA DEL SISTEMA ACTUAL .....	20
3.3    PLANTEAMIENTO DE LA SOLUCIÓN ELEGIDA .....	22
3.4    DESARROLLO DE LA SOLUCIÓN.....	23
3.4.1    Mecanismos generales para la detección de dispositivos.....	23
3.4.1.1    Bases de datos de dispositivos .....	25
3.4.1.2    WURFL.....	27
3.4.1.3    Configuración de WURFL.....	28
3.4.1.4    Rendimiento de la base de datos WURFL .....	35
3.4.2    Identificación de las categorías de dispositivos móviles.....	38
3.4.2.1    Análisis de las características más importantes de los dispositivos.....	38
3.4.2.2    Categorías de dispositivos y sus características.....	41
3.4.2.3    Tendencia de la resolución de pantalla de los dispositivos móviles.....	43
3.4.2.4    Resolución de pantalla de los dispositivos de la base de datos WURFL.....	45
3.4.3    Análisis de la página de entrada al punto de acceso Wi-Fi.....	46
3.4.3.1    Identificación de los elementos de la página.....	46
3.4.3.2    Identificación de las tecnologías de las que hace uso la página .....	48

3.4.4	Rediseño de la página web para cada categoría .....	49
3.4.4.1	<i>Diseño estático para la categoría de paisaje.....</i>	50
3.4.4.2	<i>Diseño estático para la categoría de retrato .....</i>	52
3.4.5	Adaptación automática .....	54
3.4.5.1	<i>Requisitos de la página web original.....</i>	58
3.4.5.2	<i>Atributos para controlar la futura adaptación .....</i>	60
3.4.5.3	<i>Resultado de la adaptación automática dinámica para la categoría de paisaje....</i>	63
3.4.5.4	<i>Resultado de la adaptación automática dinámica para la categoría de retrato ....</i>	65
3.4.5.5	<i>Adaptación automática de todas las páginas de la web .....</i>	67
3.4.5.6	<i>Rendimiento de la adaptación automática de la página.....</i>	68
<b>CAPÍTULO 4 - CONCLUSIONES Y TRABAJOS FUTUROS .....</b>		<b>75</b>
4.1	CONCLUSIONES .....	75
4.2	TRABAJOS FUTUROS .....	77
<b>APÉNDICE A - PRESUPUESTO.....</b>		<b>81</b>
A.1	DESCOMPOSICIÓN DEL PROYECTO EN TAREAS .....	81
A.2	COSTES DEL PROYECTO .....	82
<b>APÉNDICE B – RESULTADOS DE LA ADAPTACIÓN .....</b>		<b>84</b>
B.1	iPHONE:.....	84
B.2	NOKIA N810: .....	85
B.3	NOKIA E60:.....	86
B.4	NOKIA N61: .....	87
<b>APÉNDICE C – CÓDIGO DE LOS SCRIPT DE ADAPTACIÓN Y FUNCIONAMIENTO DE TODO EL PROCESO.....</b>		<b>88</b>
C.1	SCRIPT PARA LA DETECCIÓN DEL DISPOSITIVO .....	88
C.2	FUNCIÓN QUE ADAPTA LA PÁGINA PARA LA CATEGORÍA DE RETRATO .....	89
C.3	FUNCIÓN QUE ADAPTA LA PÁGINA PARA LA CATEGORÍA DE PAISAJE .....	90
C.4	FUNCIONES AUXILIARES PARA REALIZAR MODIFICACIONES EN LA PÁGINA.....	90
<b>APÉNDICE D – SCRIPT RENDIMIENTO WURFL .....</b>		<b>93</b>
<b>REFERENCIAS .....</b>		<b>94</b>

## Índice de figuras

Figura 1: Adaptación de la página con Google Mobilizer .....	8
Figura 2: Adaptación de la página con Mowser .....	10
Figura 3: Adaptación de la página con Skweezer .....	12
Figura 4: Mapa de los hotspots de Deutsche Telekom en Berlín .....	17
Figura 5: Aspecto de la copia de la página del punto de acceso Wi-Fi .....	19
Figura 6: Prueba de la página desde varios dispositivos .....	21
Figura 7: Prueba de la página desde un dispositivo .....	21
Figura 8: Diagrama de funcionamiento de una base de datos de terminales móviles .....	26
Figura 9: Diagrama de la estructura original de la base de datos WURFL .....	30
Figura 10: Diagrama de la estructura de la base de datos tras aplicar la corrección .....	31
Figura 11: Ejemplo de detección del dispositivo desde un Nokia N810 .....	34
Figura 12: Ejemplo de detección del dispositivo desde el navegador Firefox 3.5.8 .....	34
Figura 13: Resoluciones de pantalla típicas de los dispositivos móviles .....	43
Figura 14: Elementos de la página original .....	48
Figura 15: Imagen original y modificada manualmente con GIMP .....	51
Figura 16: Diseño estático para la categoría de paisaje .....	51
Figura 17: Diseño estático para la categoría de retrato .....	53
Figura 18: Diagrama del proceso completo de detección del dispositivo y adaptación de la página .....	57
Figura 19: Disposición de los elementos tras la adaptación automática para la categoría de paisaje .....	63
Figura 20: Resultado de la adaptación automática en un Nokia N810 (categoría de paisaje) .	64
Figura 21: Disposición de los elementos tras la adaptación automática para la categoría de retrato .....	65
Figura 22: Resultado de la adaptación automática en un Nokia E60 (categoría de retrato) ....	66
Figura 23: Diagrama de funcionamiento de .htaccess .....	68
Figura 24: Página original (izda) y página resultante de la adaptación (dcha) en un iPhone ..	84
Figura 25: Página original (arriba) y página resultante de la adaptación (abajo) en un Nokia N810 .....	85
Figura 26: Página original (izda) y página resultante de la adaptación (dcha) en un Nokia E60 .....	86

Figura 27: Página original (izda) y página resultante de la adaptación (dcha) en un Nokia N61	
.....	87

## Índice de tablas

Tabla 1: Rendimiento de la base de datos WURFL sin caché .....	36
Tabla 2: Rendimiento de la base de datos WURFL con caché .....	36
Tabla 3: Mejora en el rendimiento de la base de datos WURFL al utilizar la caché .....	37
Tabla 4: Características de las categorías .....	42
Tabla 5: Resolución de pantalla de los dispositivos utilizados en el proyecto .....	45
Tabla 6: Rendimiento de la adaptación automática (versión en PHP) .....	70
Tabla 7: Rendimiento de la adaptación automática (versión en Java) .....	72
Tabla 8: Comparativa del rendimiento de las versiones PHP y Java de la adaptación automática .....	72
Tabla 9: Presupuesto final del proyecto .....	83



## Acrónimos

**3G:** Tercera-generación de transmisión de voz y datos a través de telefonía móvil.

**Ad-hoc:** Literalmente “para esto”. En tecnología de redes se dice de las redes en las que no hay puntos de acceso centrales.

**API:** Application Programming Interface

**CC/PP :** Composite Capabilities/Preferences Profile

**CSS:** Cascading Style Sheets

**DOM:** Document Object Model

**GIMP:** GNU Image Manipulation Program

**GNU:** Acrónimo recursivo de “GNU's Not Unix!”

**GPRS:-** General Packet Radio Service

**HTML:** HyperText Markup Language

**HTTP:** HyperText Transfer Protocol

**IP:** Internet Protocol

**JSON:** JavaScript Object Notation

**PC:** Personal Computer

**PHP:** Acrónimo recursivo de “PHP Hypertext Pre-processor”

**RDF:** Resource Description Framework

**RSS:** RDF Site Summary

**UAProf:** User Agent Profile

**URL:** Uniform Resource Locator

**WAP:** Wireless Application Protocol

**Wi-Fi:** Marca que certifica que los productos que la emplean cumplen con los estándares 802.11.

**WISPr:** Wireless Internet Service Provider roaming

**WURFL:** Wireless Universal Resource FiLe

**WYSIWYG:** What You See Is What You Get

**XML:** eXtensible Markup Language

## Resumen

En la actualidad, el avance de las nuevas tecnologías ha añadido nuevas funcionalidades a los terminales móviles, para las cuales no habían sido concebidos originalmente. Un ejemplo de ello es la posibilidad de conectarse a Internet desde un dispositivo que cabe en el bolsillo del pantalón. Debido a la necesidad actual de permanecer conectados a Internet, muchos usuarios sacan provecho de sus móviles, ya sea conectándose a través de su proveedor telefónico o mediante redes Wi-Fi. El gran problema que pueden encontrar estos usuarios es que los sitios web no han sabido prepararse para la llegada de estos terminales, con unas características más reducidas que los ordenadores para los que han sido diseñados.

Este proyecto ha sido realizado en la empresa Deutsche Telekom AG, concretamente en Berlín, durante la realización de unas prácticas de seis meses de duración. El objetivo principal del proyecto era desarrollar un sistema dinámico de adaptación de la página de inicio del punto de acceso Wi-Fi para su óptima visualización en cualquier terminal móvil.

La primera parte del trabajo consistió en buscar información y analizar las distintas posibilidades que la red ofrecía en el campo de la adaptación de páginas web para terminales móviles, así como la búsqueda de distintas tecnologías de las que se podía hacer uso en la solución final. Por otro lado, la segunda parte consistió en el desarrollo propiamente dicho de la solución: una herramienta que permite la adaptación automática de la página web para su visualización en cualquier dispositivo, tras obtener las características técnicas del mismo y clasificarlo en una de las categorías definidas en el proyecto.

# Capítulo 1 - Motivación y objetivos

## 1.1 Motivación

En nuestra sociedad la tecnología avanza muy rápidamente. Como parte de la tecnología, los teléfonos móviles han visto incrementar sus características técnicas, y las tecnologías de las que hacen uso, aproximándose cada vez más a las de los ordenadores. Todo esto hace que los usuarios quieran exprimir al máximo las capacidades de sus dispositivos.

Igualmente, Internet se ha extendido muy velozmente, siendo hoy en día utilizado, no solo para obtener información, sino para comprar, para trabajar, para mantener el contacto con gente que está lejos... llegando a ser imprescindible para mucha gente, tanto en el ámbito laboral, como en el personal.

Originalmente los teléfonos móviles no pretendían que los usuarios se conectaran a Internet a través de ellos, pero pronto empezaron a utilizarse para este cometido. En esta primera etapa, los móviles eran capaces de dar acceso a páginas WAP (Wireless Application Protocol), y no a las páginas que se podían visitar con un ordenador convencional. Estas páginas WAP eran versiones sencillas de páginas web originales. Básicamente se limitaban a mostrar menús por pantalla que llevaban a pequeños fragmentos de texto y a enlaces a otras páginas WAP. Su sencillez era debida, principalmente, a dos factores:

- Las reducidas pantallas de los dispositivos, en blanco y negro, y la baja resolución de las mismas. Es fácil de imaginar la imposibilidad de mostrar una página web convencional en una pantalla en la que caben 6 ó 7 filas de texto y 2 ó 3 palabras en cada fila, como era el caso de los móviles con tecnología WAP de aquella época.
- La velocidad de conexión desde un móvil era inferior a la que se obtenía con un ordenador. Así, el acceso a una página web más compleja supondría un mayor tiempo de espera, al igual que un mayor coste para el usuario cuyas tarifas fueran en función de la cantidad de datos descargada.

La aparición de nuevas tecnologías en las redes de telefonía móvil (GPRS, 3G, etc.), junto a la mejora de las características técnicas de los terminales móviles, implicó una mejora del servicio de Internet ofrecido a los usuarios móviles. Junto a la forma tradicional de conectarse a Internet, mediante la compañía operadora con la que se tiene contratado el servicio telefónico, existe otro método, utilizando las redes inalámbricas Wi-Fi, y es que hoy en día los móviles van incorporando poco a poco antenas Wi-Fi que permiten la conexión a las citadas redes.

Algunas redes Wi-Fi son gratuitas, como las ofrecidas por algunas entidades públicas, por ejemplo el ayuntamiento de Leganés en determinados lugares públicos, con el objetivo de expandir las nuevas tecnologías. Por otra parte, también hay redes Wi-Fi de pago, que dan servicio a los clientes de apartamentos, hoteles, cafeterías, aeropuertos...

Deutsche Telekom es la mayor compañía de telecomunicaciones de Alemania. Bajo la marca T-Mobile disponen de una amplia red de puntos de acceso a Internet (*hotspots*) distribuidos por toda Alemania. Cuando un usuario desea conectarse a Internet, mediante alguno de sus puntos de acceso Wi-Fi, debe, en primer lugar, encender la antena Wi-Fi de su dispositivo y seleccionar la red T-Mobile de la lista de redes detectadas. El siguiente paso es registrarse en el punto de acceso para obtener un acceso completo a Internet, y para este cometido, T-Mobile dispone de una página desde la que, además, el usuario puede conseguir información acerca de los servicios del proveedor T-Mobile. Para llegar a dicha página, el usuario es redireccionado a la misma cuando éste intenta acceder a cualquier página web. En algunos casos, como en el de los iPhone, el dispositivo es capaz de detectar que la red Wi-Fi a la que se ha conectado es a través de un punto de acceso que requiere registro, y abre automáticamente la página del hotspot, sin necesidad de abrir el navegador e intentar acceder a cualquier web para ser redirigido.

El problema es que esta página web está adaptada y optimizada para su visionado en una pantalla de ordenador, pero el número de clientes potenciales que usan teléfonos móviles es bastante alto, debido a que la cobertura ofrecida por los puntos de acceso es principalmente en el exterior o en lugares públicos.

Desde un teléfono móvil, la página no se ve como debería, ya que en el caso de tener una pantalla y resolución pequeñas, solo se verá una mínima parte de la página teniendo que

buscar por la misma lo que se quiere. Y en caso de tener un teléfono móvil con una pantalla de calidad con una alta resolución (cercana a la de un ordenador), la página se verá demasiado pequeña como para leer el contenido, ya que a pesar de la alta resolución, el tamaño de la pantalla ni siquiera se acercará al de un monitor, al tratarse de un dispositivo de bolsillo.

En esta memoria se presenta el trabajo realizado durante la beca Erasmus Placement en Deutsche Telekom, en el departamento de investigación y desarrollo, T-Labs Berlín, de Febrero a Agosto del 2010.

Con las becas Erasmus Placement se pretende acercar al estudiante al mundo laboral, permitiéndole conocer las actividades que se llevan a cabo en los diferentes negocios e instituciones, publicas o privadas. Para conseguir esto, los estudiantes colaboran en las tareas rutinarias que son desempeñadas en dichas instituciones, siempre bajo la orientación y supervisión de un tutor. Las actividades consisten en la aplicación práctica del conocimiento, o la adquisición de habilidades prácticas.

## **1.2 Objetivos**

Durante el período de prácticas en este proyecto, se espera que se trabaje en concebir e implementar una solución para adaptar elegantemente las páginas web de entrada al punto de acceso Wi-Fi de Deutsche Telekom para los dispositivos móviles. Este trabajo se estructura en dos partes:

- La primera parte incluye un análisis profundo del problema, y debe resultar en una visión general de las posibles soluciones, sus pros y sus contras, para dar una propuesta de la solución preferida.
- La segunda parte del trabajo consiste en la implementación de la solución propuesta. Esta implementación se espera que sea un prototipo, pero que esté orientada al entorno del sistema real tanto como sea posible.

Un requisito imprescindible es que, para la adaptación de las páginas web, se debe ser mínimamente intrusivo, entendiendo con esto que, en la medida de lo posible, no se debe modificar la página web original. No obstante, podría ser aceptable el ofrecer unas recomendaciones para el diseño y estructura de la página original, siempre que estas estén

orientadas a facilitar la posterior adaptación para los dispositivos móviles, así como a mejorar la estructura original de la página y corregir fallos. En cualquier caso, un objetivo prioritario en este proyecto es mantener la página web original intacta, trabajando en una solución que, a partir de la página original, realice la adaptación para los dispositivos móviles.

Para llegar a cumplir estos objetivos, el proyecto se ha dividido en las siguientes tareas:

- Análisis del problema, incluyendo partes teóricas y prácticas.
- Investigación del Estado del Arte en el campo de la adaptación de sitios web.
- Desarrollo de un prototipo de solución basado en el análisis del problema y el Estado del Arte.
- Presentación de este prototipo a un pequeño auditorio de expertos, y posterior discusión.
- Implementación de la solución en un entorno de laboratorio.
- Comprobación real con un conjunto de dispositivos móviles.
- Estimación / descripción de las tareas requeridas para la migración de la solución planteada al sistema real de un punto de acceso T-Mobile.
- Presentación final de actividades.

El trabajo será constantemente monitorizado por un tutor en la empresa. Se realizarán presentaciones a otros expertos de Deutsche Telekom Laboratories para conocer otros puntos de vista durante la realización del trabajo. Por otra parte, el trabajo se orientará a los requisitos del entorno del sistema real, y los resultados serán comunicados al departamento encargado del Hotspot de Deutsche Telekom.

## Capítulo 2 - Estado del Arte

En la actualidad, las compañías con presencia en la red se encuentran con un problema a la hora de definir los usuarios potenciales de sus sitios web, ya que cualquiera, desde cualquier dispositivo con acceso a Internet, puede visitarles, y están en juego muchos factores: su imagen de marca, la fidelidad del usuario... por lo que deben estar preparados para todo con el fin de que la impresión del usuario sea la mejor posible.

Partiendo de esta premisa, las compañías se pueden plantear originalmente la idea de tener una página web principal, orientada a los usuarios de ordenadores, y una versión móvil de dicha web, para los usuarios de dispositivos móviles. Esto que puede considerarse un avance, no es una solución óptima, puesto que la versión móvil es siempre la misma para todos los dispositivos móviles, sin tener en cuenta las características técnicas de los mismos, por lo que podría suceder que los usuarios de móviles de características medio-altas se sientan frustrados con las versiones móviles de las páginas, que son demasiado básicas para ellos, y con las páginas originales, que son demasiado complejas.

La solución ideal sería aquella en la que, la página que esta viendo un usuario en su dispositivo, se ajuste de tal forma al mismo que el propio usuario piense que ha sido diseñada expresamente para él. Esta línea siguen algunas compañías con sus páginas web, pero únicamente para algunos modelos concretos de dispositivos, como por ejemplo para el popular iPhone. En estos casos la experiencia de usuario obtenida desde un iPhone es obviamente inmejorable, pero el problema sigue estando en el resto de dispositivos del mercado, ya que se les sigue ofreciendo una versión móvil idéntica para todos.

A pesar de todo, estas soluciones aportadas por las compañías en la red, están generalmente basadas en realizar distintos diseños de su página web original manualmente, orientados cada uno a un determinado usuario. Otra opción realmente interesante es la de obtener una adaptación automática del sitio web principal, sin que signifique el diseñar desde cero una segunda página web, y es en este punto en el que se centra más este proyecto. Por esto, se ha creído conveniente realizar un estudio pormenorizado sobre las posibilidades existentes en el campo de las adaptaciones automáticas, con sus pros y sus contras.

## **2.1 Visión general de las soluciones de adaptación existentes**

Hoy en día existen varios transcodificadores HTML (del inglés transcoder, también llamados movilizador, del inglés mobilizer), como son llamadas las aplicaciones (generalmente un servicio on-line en Internet) que, dado un sitio web, hace una modificación de la página para mejorar su usabilidad en las pequeñas pantallas de un dispositivo móvil, normalmente con peores capacidades técnicas que un ordenador de sobremesa. En otras palabras, estas aplicaciones generan una versión “móvil” de un sitio web disponible en la red. Muchos sitios web intentaron mantenerse ofreciendo este servicio, pero no todos siguen en activo, como se ha podido detectar después del estudio realizado.

### **2.1.1 Panorámica inicial**

Estos transcodificadores podrían ser clasificados en dos categorías, de acuerdo al tipo de sitios web que son capaces de tratar:

- Sitios con redifusión web (RSS): Algunos de estos transcodificadores necesitan la URL que algunos sitios web ofrecen para su redifusión web, y no la URL del sitio web normal, para crear una versión adaptada para los móviles de la página original. Por esta razón, este tipo de transcodificadores son usados para leer, en un formato conveniente cuando se utiliza un dispositivo móvil, sobre todo, sitios de noticias, blogs, etc. Algunos ejemplos de este tipo de servicio son [www.mob-it.com](http://www.mob-it.com) y [www.feedm8.com](http://www.feedm8.com). Sin embargo hay una serie de problemas al usar solamente la URL de RSS para crear un sitio móvil, y es que alguna información de las páginas se perdería, ya que no es la página web completa la que se trata, lo que hace estos transcodificadores no muy certeros, e incluso inútiles si la página web que se quiere adaptar no cuenta con servicio RSS, ya que en ese caso no se puede adaptar.
- Cualquier sitio web: Algunos transcodificadores toman directamente una página HTML y la adaptan para hacerlas que se muestren correctamente en un teléfono móvil. Éstos reformatean las páginas HTML, modifican el tamaño de las imágenes y cambian el formato de los textos, todo ello de una forma muy predecible y lineal. Este tipo de adaptación podría ser útil para ver la apariencia de la página web del punto de



acceso después de ser adaptada por este tipo de aplicaciones, y así poder analizar las modificaciones que ellos efectúan.

### 2.1.2 Transcodificadores generales

Como se dijo anteriormente, muchos transcodificadores no han tenido éxito, por lo que los sitios webs más importantes que continúan ofreciendo este servicio, y que han sido analizados, son:

- Google Mobilizer: Google, desde su situación de líder en algunos servicios por Internet, intentó satisfacer a los usuarios móviles ofreciendo una herramienta que adapta automáticamente cualquier sitio en la red a una versión móvil del mismo, con el fin de atraer más usuarios a Google. En principio, para Google es más fácil que para otras compañías introducir un nuevo servicio como éste, puesto que los usuarios ya conocen la marca Google, e incluso confían en ella.
- Mowser: Mowser pertenece a dotMobi, una compañía líder en servicios por Internet móvil, lo que convierte a Mowser en una herramienta bastante confiable, al igual que Google Mobilizer. DotMobi también ofrece también una herramienta muy similar llamada Instant Mobilizer, pero en este caso, para conseguir la versión adaptada automáticamente de un sitio web, es necesario, primero, elegir uno de sus socios con los que obtener un dominio en Internet, lo que significa que hay que pagar a otra compañía por el dominio en el que se hospedará el sitio web adaptado. Asimismo, dotMobi es la compañía propietaria de DeviceAtlas, una de las bases de datos de dispositivos móviles que se estudiará y explicará posteriormente durante la realización de este proyecto.
- Skweezer: Skweezer no es solo un movilizador fiable, sino también un motor de búsqueda para móviles, un directorio de sitios web para móviles, e incluye otros servicios ofrecidos para los usuarios registrados. De acuerdo a varios sitios web y blogs, Skweezer ha sido el transcodificador líder últimamente, y una prueba de ello son los premios que ha Ganado los últimos años, por ejemplo, consiguiendo el "Gold Star" en la categoría "Personal Software: Mobile Web Content Utility" de los "Mobile Star Awards" organizados por MobileVillage en los años 2005 y 2006, y siendo el ganador en la categoría "Pocket PC: Web Compression Service" de los prestigiosos "Best Software Awards" de la revista "Smartphone & Pocket PC magazine" en los años 2007 y 2008.

### 2.1.2.1 Google Mobilizer

En la figura 1 se puede ver el sitio web del hotspot de T-Mobile tras ser adaptado por Google.

This page adapted for your browser comes from thoralfbrandt-arbeit.dyndns.org and is not endorsed by Google.  
Page shown from here. [Go to page 1](#)

**Startseite - Login**

[HotSpot Kunde werden](#)

[Tarife](#)

[HotSpot Standorte & Partner](#)

[Sicherheit](#)

[Hilfe](#)

[Download](#)

[My HotSpot](#)

[Kontakt](#)

HotSpot Kunde werden

HotSpot Zugang **ohne Vertragsbindung** kaufen  
HotSpot Pass 60 Minuten für 8,00 Euro  
HotSpot Pass 30 Tage für 29,00 Euro

HotSpot Pass kaufen (Kreditkarte) >

HotSpot Pass kaufen (T-Pay) >

HotSpot Zugang **mit Vertragsbindung** kaufen

HotSpot Tarife für T-Home, T-Mobile  
und DSL-Business Kunden >

## Bereits HotSpot Kunde?

Benutzername


Hilfe zur Eingabe

Passwort

[Passwort vergessen?](#)

Ich akzeptiere die AGB und  
Zahlungsbedingungen zu HotSpot.

## Roaming

 Wireless  
Broadband Alliance

T-Home & T-Mobile © 2009 | [Kontakt](#) | [AGB](#) | [Impressum](#)

[Back to search results](#)  
[Google Home](#)

Formatted for mobile viewing by Google

Figura 1: Adaptación de la página con Google Mobilizer

Se puede observar que el formato original HTML ha sido completamente eliminado y todos los elementos son mostrados en una columna, empezando por el menú. Solo las imágenes que son enlaces permanecen formateadas con los típicos colores de T-Mobile rosa y blanco. Sin embargo, las imágenes que no son enlaces son completamente eliminadas en la versión adaptada. El formato de las fuentes parece seguir siendo el mismo que antes. Los elementos del menú son mostrados uno detrás de otro, pero dejando demasiado espacio entre los enlaces, dando un nuevo formato a cada enlace con una fuente subrayada de color azul. Los elementos de listas son mostrados en filas diferentes, pero los símbolos que identifican el principio de cada elemento son eliminados, por lo que los usuarios no serían capaces de reconocer estos elementos como una lista.

Sobre todas estas modificaciones, hay un detalle que hace que la versión adaptada del sitio web no sea utilizable en absoluto: las cajas de texto en las que el usuario debe escribir su nombre de usuario y su contraseña han desaparecido, al igual que la casilla de verificación que tiene que ser marcada para aceptar los términos y las condiciones del contrato. También ha desaparecido la lista desplegable para elegir compañía telefónica del apartado roaming. Por todo esto se puede asumir que Google Mobilizer no puede tratar convenientemente algunos elementos de formularios HTML, lo que no es aceptable.

A pesar de tener que adaptarse a una pantalla pequeña, ya que se supone que la página tiene que ser mostrada en un dispositivo móvil, Google Mobilizer añade una cabecera, al principio de la página web, para hacer saber al usuario la URL de donde viene la página original, e informar de que la página web no ha sido aprobada por Google. También añaden un pie de página, al final de la misma, donde sitúan el motor de búsqueda Google, y advierten que la página mostrada no es la original, sino una versión adaptada para su visualización en terminales móviles.



#### **2.1.2.2 Mowser**

El segundo transcodificador probado ha sido Mowser, y en la figura 2 se puede apreciar como se ve la página después del proceso de adaptación.


dotMobi
mobiThinking
mobiForge
ready.mobi
DeviceAtlas
find.mobi
Mowser

[English](#) | [Kontakt](#) | [AGB](#) | [Impressum](#)  
[Startseite - Login](#)  
[HotSpot Kunde werden](#)  
[Tarife](#) [HotSpot Standorte & Partner](#) [Sicherheit](#) [Hilfe](#) [Download](#) [My HotSpot](#) [Kontakt](#)  
[HotSpot Kunde werden](#)

HotSpot Zugang **ohne Vertragsbindung** kaufen

HotSpot Pass 60 Minuten für 8,00 Euro   HotSpot Pass 30 Tage für 29,00 Euro  
[HotSpot Pass kaufen \(Kreditkarte\)](#)     
[HotSpot Pass kaufen \(T-Pay\)](#)   

HotSpot Zugang **mit Vertragsbindung** kaufen

HotSpot Tarife für T-Mobile und DSL-Business Kunden     
[\[ https://hotspot.handelsblatt.com/tho \]](https://hotspot.handelsblatt.com/tho)  
[Bereits HotSpot Kunde?](#)

Benutzername



[Hilfe zur Eingabe](#)

Passwort

[Passwort vergessen?](#)

☐ Ich akzeptiere die [AGB](#) und [Zahlungsbedingungen](#) zu HotSpot.

Login

Roaming  
  
Please choose... 

T-Home & T-Mobile © 2009 | [Kontakt](#) | [AGB](#) | [Impressum](#)

You are viewing a mobilized version of this site...  
[View original page here](#)


Mobilized by  Mowser

Figura 2: Adaptación de la página con Mowser

Con este movilizador se obtiene un mejor resultado que con Google Mobilizer. En este caso el usuario puede encontrar los campos del nombre de usuario y contraseña, por lo que al menos el sitio web tendría funcionalidad. De igual forma, los elementos del menú son mostrados de una mejor manera que con la versión adaptada por Google Mobilizer. Así, aunque los elementos del menú aparecen todos en la misma fila, y sin ningún tipo de formato, aparte del color azul de la fuente, al menos no ocupan demasiado espacio.

Los elementos de las listas HTML son todos mostrados en una fila también. Y sin los símbolos demarcadores entre elemento y elemento que hacen saber al usuario que dichos elementos pertenecen a una lista. Un punto a favor de Mowser es que las imágenes no son eliminadas, incluso aquellas que no son enlaces, pero, por otra parte, el mayor problema con Mowser es que varios caracteres especiales, como "€" o "ü", no aparecen correctamente. Aparte de este asunto, algunas imágenes, las que son enlaces y tienen algún texto, no son adaptadas correctamente: Mowser modifica su tamaño, pero haciéndolas demasiado pequeñas, por lo que el texto es demasiado pequeño, lo que dificulta su lectura.

Mowser completa esta adaptación anunciándose a sí mismo añadiendo una cabecera y un pie de página, al principio y al final de la página adaptada, respectivamente. En la cabecera, Mowser incluye varios enlaces a diferentes servicios en los que la compañía que desarrolla Mowser, dotMobi, está involucrada, como por ejemplo, DeviceAtlas, una base de datos de dispositivos que ya ha sido mencionada anteriormente. En el pie de página, Mowser recuerda al usuario que el sitio web actual es una versión adaptada, movilizada por Mowser, y añade un enlace al sitio web original.

### **2.1.2.3 Skweezer**

El último transcodificador probado ha sido Skweezer. En la figura 3 se muestra las pruebas realizadas con él para conseguir una versión adaptada del sitio web, y poder realizar posteriormente un análisis de ello.



[Play MOBILE Wars On ur Firefox](#)

[English](#) | [Kontakt](#) | [AGB](#) | [Impressum](#)

[Startseite - Login](#)

[HotSpot Kunde werden](#)

, [Tarife](#), [HotSpot Standorte & Partner](#), [Sicherheit](#), [Hilfe](#), [Download](#), [My HotSpot](#), [Kontakt](#)

## HotSpot Kunde werden

HotSpot Zugang **ohne Vertragsbindung** kaufen

HotSpot Pass 60 Minuten für 8,00 Euro HotSpot Pass 30 Tage für 29,00 Euro,

[HotSpot Pass kaufen \(Kreditkarte\)](#) >

[HotSpot Pass kaufen \(T-Pay\)](#) >

HotSpot Zugang **mit Vertragsbindung** kaufen

[HotSpot Tarife für T-Home, T-Mobile und DSL-Business Kunden](#) >

09:57 Uhr

[Merck's Milliardendeal vergault die Aktionäre](#)

10:14 Uhr

[Goldman-Swap gehört nun einer Briefkastenfirma](#)

09:30 Uhr

[i-Flow kündigt nächste Sonata-Generation an](#)

07:22 Uhr

[Die zwiespältigen Winterspiele sind beendet](#)

Schlagzeilen des Tages aus der Handelsblatt.com-Redaktion

## Bereits HotSpot Kunde?

Benutzername

Hilfe zur Eingabe

Passwort

[Passwort vergessen?](#)

☐ Ich akzeptiere die AGB und Zahlungsbedingungen zu HotSpot.

Login

## Roaming



Please choose... ▼

Submit Query

T-Home & T-Mobile © 2009 | [Kontakt](#) | [AGB](#) | [Impressum](#)

**skweezer®**

58% (23.81 kB) saved

Figura 3: Adaptación de la página con Skweezer

Como era de esperar, debido a los premios que Skweezer había conseguido, el resultado es bastante bueno. Como con Mowser, los elementos del menú aparecen todos en la misma fila, lo que ahorra espacio. En este caso, Skweezer da a los elementos un formato característico: la fuente es verde y subrayada. Los elementos de las listas aparecen también todos en la misma fila, sin el símbolo típico que define cada elemento de la lista. El formato de las fuentes parece ser el mismo que en la página original, y Skweezer respeta las imágenes originales, incluso aquellas que son enlaces, pero, por alguna razón desconocida, la imagen que es un enlace en la sección roaming no es mostrada correctamente. Esto podría considerarse como una excepción, ya que tiene el mismo formato que otras imágenes que son enlaces y sí son mostradas correctamente.

Las cajas de texto donde los usuarios tienen que escribir el nombre de usuario y la contraseña están correctamente adaptadas, e incluso el espacio que ocupan no es tanto como en la página adaptada con Mowser. Curiosamente, la lista desplegable para elegir compañía telefónica en el apartado roaming, cambia su comportamiento después de ser adaptado. Skweezer añade un botón con la leyenda “Submit Query” que tiene que ser presionado para ser redirigido a la página web de la compañía roaming seleccionada, después de haberla elegido en la lista desplegable. Esta redirección se hace automáticamente en la página original, así como en la página adaptada por Mowser, por lo que parece que Skweezer quiere hacer el funcionamiento de la lista desplegable mas fácil de ejecutar para dispositivos con capacidades técnicas más bajas pero, por otra parte, tanto el diseño como la usabilidad de la página cambian considerablemente. Por lo tanto, dependiendo de lo que se requiera, esta modificación que realiza Skweezer puede considerarse un defecto o una virtud.

Se puede apreciar que Skweezer ha sido el único transcodificador que ha intentado adaptar la sección de noticias. Tanto Google como Mowser simplemente ignoraron esta sección eliminándola. Esta sección es un marco en el que aparecen noticias desplazándose continuamente de abajo a arriba del mismo. Skweezer, durante el proceso de adaptación, escribe todas las noticias una detrás de otra. Esto es también algo que depende de lo que se requiere inicialmente para poder evaluar si es mejor realizar esta adaptación, o directamente eliminarlo como hacen los otros transcodificadores.

Un detalle que debería ser observado es que Skweezer modifica el formato de todos los enlaces en la página adaptada dándoles su color corporativo, verde, por lo que la

apariciencia de la página cambia completamente, y la impresión producida por el sitio web es absolutamente diferente, comparándolo con la página original.

Como el resto de movilizadores, Skweezer añade una cabecera al principio de la página adaptada, aunque en este caso, sitúan publicidad externa en el mismo. Hay también un pie de página al final de la misma, con el logotipo de Skweezer, y en él dan información acerca de cuanto espacio, en Kilobytes, se han ahorrado gracias a haber hecho la adaptación. Además, en el pie de página, aparecen algunos enlaces relacionados con Skweezer, y otro enlace para ser redireccionado al sitio web original, sin ser adaptado.

### 2.1.3 Resultados generales de los transcodificadores

Después de este estudio con 3 transcodificadores de los más conocidos, se pueden extraer algunas conclusiones acerca de los movilizadores o transcodificadores:

- Utilizarlos es un proceso muy cómodo, en el sentido de que es un proceso completamente automático.
- No todas las páginas mantienen su funcionalidad después del proceso de adaptación, como en el caso de Google Mobilizer, con el que las cajas de texto, como las del nombre de usuario y contraseña, son eliminadas. Este es un punto crucial, ya que es inútil tener un sitio web fácil de usar, de buena apariencia y adaptado a los móviles, pero sin ninguna funcionalidad en absoluto.
- La página adaptada es la misma para cualquier dispositivo. Los transcodificadores no detectan el dispositivo, no les importan sus características. Ellos simplemente proponen una versión movilizada del sitio web original, la misma versión para un dispositivo antiguo que para el último modelo.
- La semántica de la página es absolutamente ignorada. Los transcodificadores no tienen en cuenta si un elemento del sitio web es muy importante y tiene que estar en cierta posición, o si otro elemento no es tan importante para un usuario móvil y podría ser eliminado. Este punto es bastante importante, ya que los usuarios de ordenador tienen requisitos distintos a los de los usuarios móviles. Los primeros pueden ver en sus grandes pantallas varios elementos de los sitios web, de hecho, están diseñados para ellos. Por otra parte, los usuarios móviles deberían desplazarse verticalmente, e incluso horizontalmente, por la pantalla para ver todos los elementos en una página web normal, por lo que es necesario suministrarles un sitio web alternativo, una



adaptación inteligente con la que puedan encontrar lo que quieran fácilmente, y los transcodificadores no pueden ofrecérselo.

- Todos los transcodificadores que han sido probados, modifican el diseño de la página original, por ejemplo, dando a los enlaces su color corporativo, o añadiendo publicidad en la cabecera o en el pie de página. Esto no es lo deseable en una página adaptada para los móviles, donde su diseño debería hacer recordar al usuario el diseño del sitio original.
- Algunos detalles menores son: JavaScript no es completamente soportado por los transcodificadores, por lo que algunos scripts no son tratados en la versión móvil. Las imágenes son también un problema para los transcodificadores, ya que no son correctamente adaptadas. Por ejemplo, sus tamaños son demasiado pequeños para leer el texto que incluyen, e incluso algunas imágenes, que son enlaces, directamente desaparecen.

Por todas estas razones, queda probado que los transcodificadores no son una opción profesional para llevar a cabo la tarea de adaptar un sitio web para su visualización en terminales móviles. No obstante, es una buena práctica el analizar cómo estas herramientas automáticas realizan este proceso, para comprender sus ventajas, y considerar los fallos que cometen, y ser capaces de pensar como mejorarlo en la propuesta final.

#### **2.1.4 Otras formas de adaptación automática**

Aparte de este tipo de transcodificadores, algunos navegadores de Internet incorporan una funcionalidad que consiste en intentar mejorar la experiencia de usuario, cuando se navega por la red desde un dispositivo móvil, realizando adaptaciones al vuelo de las páginas que se visitan. Sin embargo, esto queda fuera del objetivo de este proyecto, ya que los usuarios del punto de acceso de T-Mobile podrían estar utilizando cualquier navegador en sus teléfonos móviles, y no puede ser obligatorio el uso de un navegador específico para registrarse en el punto de acceso. Opera Mini es un navegador que es capaz de realizar algún tipo de adaptación para los dispositivos móviles exitosamente. Otro ejemplo es Teashark, otro navegador que hace algo similar a lo que Opera Mini realiza.

También debería ser mencionado que, los dispositivos que fueron suministrados por la compañía durante el desarrollo del proyecto, cuyo sistema operativo es Windows Mobile,

mostraban por defecto una versión modificada de cualquier sitio web que se visitara. Esto es una opción que se puede desactivar desde el menú del navegador, para evitar que éste realice alguna modificación a las páginas, y poder verlas con su diseño original, como si el dispositivo fuera un ordenador de sobremesa. Este hecho es importante, ya que estos cambios automáticos podrían ser hechos a una versión ya adaptada del sitio web, lo que podría echar a perder la estructura de la versión adaptada y hacer esta adaptación inútil.

## Capítulo 3 - Análisis, diseño y desarrollo.

### 3.1 Pasos previos necesarios

#### 3.1.1 Copia local del punto de acceso

En primer lugar, se hace necesario obtener una copia de la página web de entrada al punto de acceso Wi-Fi, con la que poder trabajar localmente.

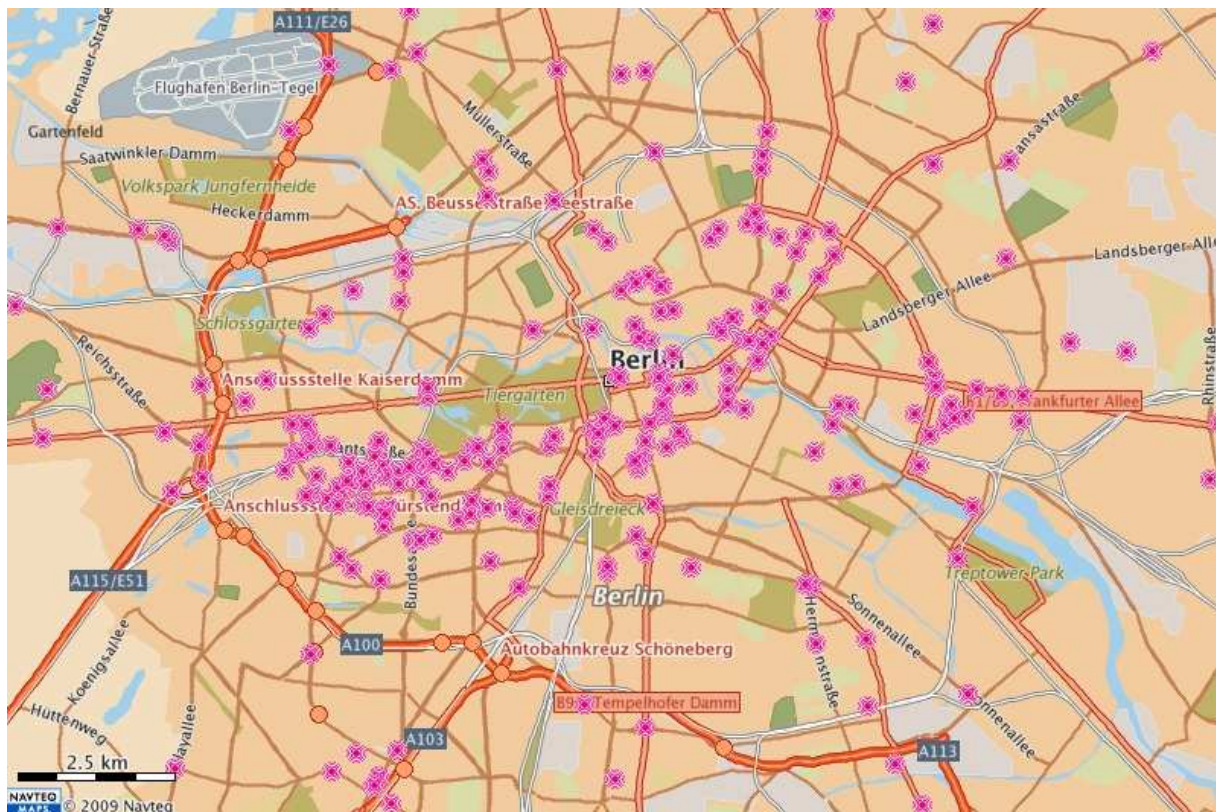


Figura 4: Mapa de los hotspots de Deutsche Telekom en Berlín

Para conseguir una copia de las páginas web que permiten conectarse al punto de acceso Wi-Fi hay que acercarse a un punto de acceso de T-Mobile con un portátil y comprobar la disponibilidad de la conexión. En la figura 4 se puede ver un plano de Berlín en el que se puede observar la localización de los puntos de acceso a Internet que la compañía tiene repartidos por la ciudad. Una vez que el portátil se ha conectado a la red ofrecida por el punto de acceso, se requiere utilizar “wget”, una herramienta de código abierto perteneciente

al proyecto GNU. La sentencia apropiada, con sus argumentos necesarios, que ha sido utilizada para conseguir una copia local de las páginas del punto de acceso inalámbrico es la siguiente:

```
wget --mirror -w 2 -p --html-extension --convert-links -P //<dir>// //<HTML_website>//
```

A continuación se explica la sintaxis de la sentencia, detallando el significado de cada uno de los argumentos especificados en la misma:

- "--mirror" : define la descarga como recursiva y establece la profundidad de dicha recursividad a infinita, teniendo en cuenta que no se obtendrán otra vez ficheros que son más antiguos que los que ya están presentes en la máquina local.
- "-w 2" : establece en 2 segundos el tiempo que se esperará entre recuperación y recuperación, para aligerar la carga del servidor.
- "-p" : declara que se tendrá que obtener todo lo necesario para mostrar páginas HTML completamente, como por ejemplo imágenes.
- "--html-extension" : declara que los documentos HTML serán guardados con la extensión ".html".
- "--convert-links" : define la modificación de todos los enlaces del documento HTML descargado, haciéndolos que apunten a los ficheros locales, en lugar de referenciar a direcciones web.
- "-P //<dir>/" : define la ruta donde los ficheros van a ser guardados.
- "//<HTML\_website>/" : define la URL de la que se debe hacer la copia local.

Después de que el comando “wget” termine de copiar todas las páginas, debería ser posible visitar la página web localmente, accediendo a la copia realizada, sin que sea necesaria la conexión a Internet o al punto de acceso. Si es así, todo está preparado para configurar un servidor web que almacene la página que acaba de ser copiada, con la finalidad de hacerla disponible para distintos dispositivos móviles.

En la figura 5 se puede ver la página de entrada del punto de acceso Wi-Fi de T-Mobile en un navegador web de escritorio, en concreto Google Chrome. El resultado es bastante bueno. Todo está donde debería estar, bien organizado, y la información expuesta es clara y precisa, haciendo al usuario sentirse cómodo con el diseño del sitio web.

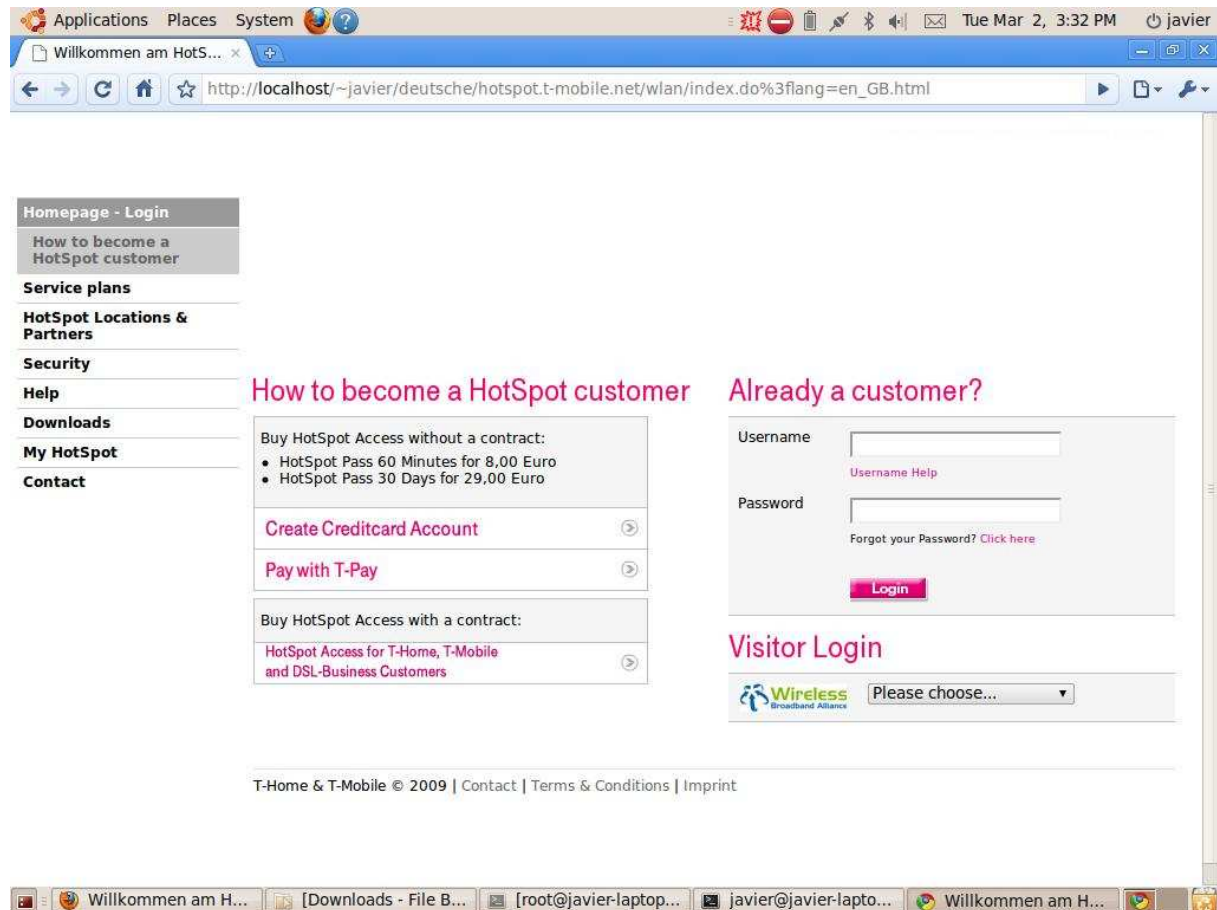


Figura 5: Aspecto de la copia de la página del punto de acceso Wi-Fi

Más tarde, esta página web será analizada para identificar la prioridad de sus elementos para los usuarios móviles, y para conocer las tecnologías de las que hace uso, con el fin de determinar si los dispositivos móviles van a ser capaces de admitirlas o no.

### 3.1.2 Preparación de un servidor web que almacene la copia de la página

Para poder trabajar con la copia local de la página web recién obtenida, hay que configurar una máquina, en este caso un ordenador portátil, como servidor web, para posteriormente almacenar la copia de la página en el mismo.

El servidor web que ha sido usado para almacenar la copia del sitio web del punto de acceso inalámbrico es un servidor “Apache” Versión 2.0, ejecutado bajo Ubuntu 9.10. Una vez que el servidor Apache fue correctamente configurado, y que los ficheros que constituyen la copia del sitio web fueron situados en la carpeta “*public\_html*”, fue posible navegar por la página web, accediendo a una ruta del estilo “*http://localhost/~javier/...*”, para comprobar si el servidor Apache estaba correctamente instalado y configurado.

Posteriormente, cualquier dispositivo móvil podía acceder a la copia local de la página web almacenada en el servidor Apache, escribiendo la dirección IP de la máquina local que hacía de servidor, con la única condición de que el dispositivo estuviera conectado a la misma subred. En este caso la dirección web era del siguiente estilo: "http://192.168.178.12/~javier/...".

Debido a que algunos dispositivos antiguos no eran capaces de conectarse a la red Wi-Fi disponible en la oficina en la que se desarrollaba el proyecto, fue necesario compartir la conexión del ordenador portátil mediante una conexión ad-hoc con ellos, para poder acceder al sitio web duplicado almacenado en el servidor.

Además, fue necesario publicar la copia del sitio web en la red para hacerla disponible a algunos proxies de Internet (transcodificadores) que generan automáticamente una versión para móviles de cualquier sitio web. Estos proxies no son capaces de obtener la página web que tienen que convertir, si ésta no está en Internet, pero esta tarea ya ha sido explicada en el capítulo Estado del Arte de esta memoria.

### **3.2 Prueba del sistema actual**

Después de que la empresa en la que se realiza el proyecto suministrara varios dispositivos móviles, se han llevado a cabo varias pruebas para comprobar como se ve la página web de entrada al punto de acceso Wi-Fi. La experiencia difiere dependiendo del modelo que está siendo utilizado, pero en todos ellos se puede observar que la página web no es tan fácil de usar como cuando se hace la prueba desde un ordenador de sobremesa. E incluso en algunos terminales, aquellos con las peores capacidades técnicas, el sitio web no es utilizable en absoluto, ya que el tamaño de la página es muchas veces más grande que el tamaño de la pantalla, consiguiendo como resultado que el usuario se pierda en el gran contenido. En las imágenes de las figuras 6 y 7 aparecen algunos ejemplos de como se ve la página en varios dispositivos móviles.



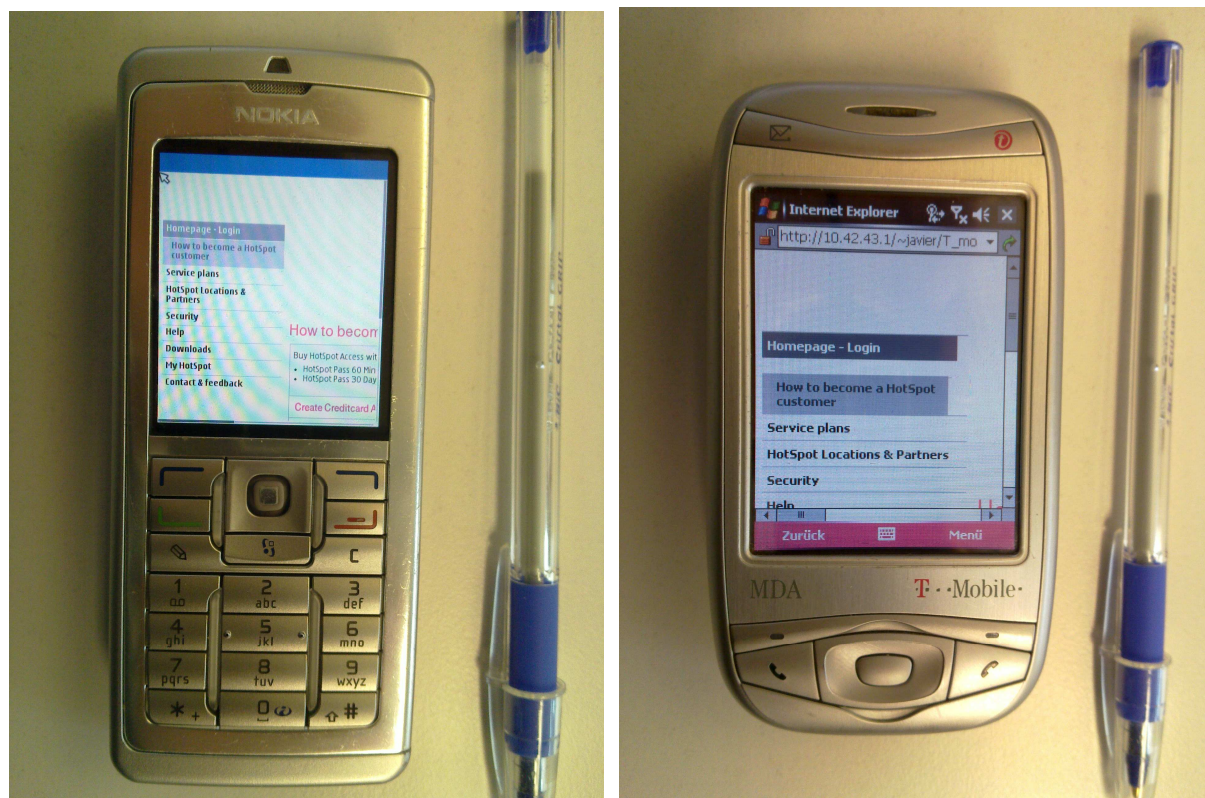


Figura 6: Prueba de la página desde varios dispositivos

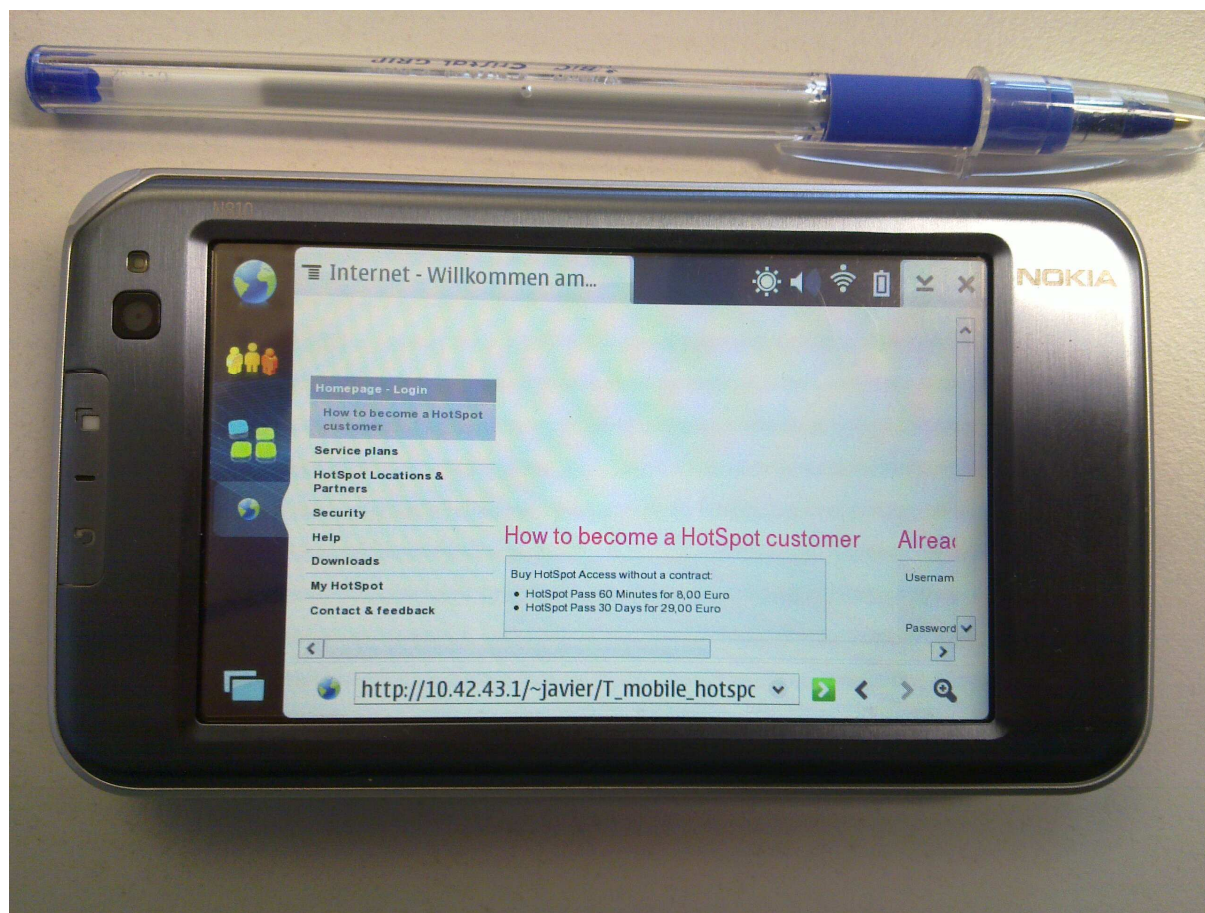


Figura 7: Prueba de la página desde un dispositivo

Como puede verse, en ninguno de ellos, ni siquiera en el que tiene mejores características técnicas, el Nokia N810, la localización de las cajas de texto en las que el usuario debería introducir el nombre de usuario y la contraseña se muestran cuando se carga la página, lo que obliga al usuario a buscarlas por toda la página. Esto no debería suceder, ya que el objetivo principal de la esta página es realmente iniciar sesión en el punto de acceso. Así queda demostrado que la estructura de la página no se ajusta correctamente a los dispositivos móviles, no es cómoda para los usuarios en absoluto, lo que todo ello hace a la página web no utilizable para el usuario de dispositivos móviles.

### **3.3 Planteamiento de la solución elegida**

Después del estudio acerca de las diferentes posibilidades que existen en la actualidad a la hora de la adaptación automática de sitios web, y que ha sido detallado en el capítulo Estado del Arte de la presente memoria, ha quedado demostrado que, en la solución ideal, el usuario debe percibir que la página se ajusta perfectamente a las características de su dispositivo. Para acercarse a este objetivo, se hace necesario obtener de alguna forma las características del dispositivo, para poder trabajar en adaptar la página web de acuerdo a dichas características.

Una vez que se conocen las capacidades del dispositivo, se podría optar por adaptar la página web al mismo, pero esto plantea unos problemas, como por ejemplo que la cantidad de dispositivos diferentes a tratar es desconocida, puesto que en el futuro las compañías diseñaran nuevos terminales con características distintas, con lo que el hacer la adaptación única para cada dispositivo plantearía un trabajo y un gasto excesivo, además de no poder abarcar a los futuros móviles por desconocimiento de sus características en momento actual, por lo que la solución no sería óptima, al haber siempre móviles sin su correspondiente página adaptada.

Por todas estas razones, para la realización del proyecto, se opta por clasificar todos los dispositivos actuales en categorías de móviles de características similares. Así se acotan un poco todas las variaciones posibles de características, pero los usuarios siguen teniendo una buena experiencia de usuario, porque, aún en el caso de tener un número determinado de categorías se puede trabajar para dotar a las páginas adaptadas de un comportamiento



dinámico, el cual se explicará más adelante. De esta forma, la adaptación es dinámica, con lo que el usuario puede observar que la página servida para su dispositivo se ajusta perfectamente, e incluso parece adaptada expresamente para él.

Tras definir las categorías en las que se clasificarán todos los dispositivos, hay que rediseñar la página web original para cada una de las categorías. Para esto hay que analizar la página, y así poder determinar la importancia de los elementos de la página y definir prioridades en los mismos, con el fin de saber como recolocarlos para cada categoría, e incluso saber si se puede eliminar alguno para una determinada categoría en concreto. En este paso, se obtendrá una página adaptada para cada categoría, pero dicha página será estática, es decir, la página será igual para todos los dispositivos de la categoría. Por eso, el siguiente paso es dotar a la página del comportamiento dinámico.

Por otra parte, cabe destacar que, en la realización del proyecto, se ha optado por desarrollar una herramienta que permita realizar la adaptación de la página al vuelo, es decir, en el momento exacto en que el navegador del dispositivo hace la petición de la página al servidor, en lugar de simplemente tener la página ya adaptada para cada categoría guardada en el servidor y redirigir a los dispositivos. Esto se ha hecho así, sin redirecciones, por su versatilidad, ya que en cualquier momento se podrían obtener las páginas adaptadas para cada categoría, si se quisiera tenerlas en memoria, en lugar de realizar la adaptación al vuelo. Además hay que tener en cuenta que, durante la realización del proyecto, no se conocían las características específicas del entorno real de los puntos de acceso Wi-Fi, ni se tenían unas directrices, en forma de requisitos, por parte del departamento que los controla, por lo que de esta forma se estaba preparado para cualquier criterio que pudiera surgir posteriormente.

### **3.4 Desarrollo de la solución**

#### **3.4.1 Mecanismos generales para la detección de dispositivos**

Hay diferentes formas de averiguar las características del navegador, o del dispositivo, que esta accediendo a un sitio web. Por ejemplo, hay un objeto JavaScript, “window.screen”, el cual contiene información básica acerca de la pantalla del dispositivo, como la resolución de la pantalla. Con esta información, se pueden diseñar páginas web para que se ajusten a las particularidades de cada pantalla. Sin embargo, esta forma de conseguir la información del

dispositivo es realizada demasiado tarde, ya que requiere una interacción con el cliente, aparte del hecho de que no todos los dispositivos móviles que pueden acceder a Internet tienen por qué admitir JavaScript. Es por eso que esta opción no ha sido elegida para la realización del proyecto.

Hay dos formas principales de llevar a cabo la detección de las características de un dispositivo en HTML:

- Usando cabeceras HTTP: los dispositivos móviles envían una cabecera adicional en el paquete HTTP – una cabecera perfil, específica de los terminales móviles. Este perfil es una URL que direcciona a un fichero RDF, el cual incluye detalles relacionados con la resolución de pantalla, particularidades del dispositivo, etc. El fichero RDF está en un formato estándar, por lo que, para todos los dispositivos, sus ficheros RDF asociados van a tener la misma estructura. Sin embargo, uno de los problemas de esta aproximación, es el hecho de que podría haber dispositivos que no se anuncien a sí mismos enviando el enlace al fichero RDF en el paquete HTTP, y en este caso se necesitaría una forma adicional de obtener las características del aparato. Por lo tanto, contar con los ficheros RDF para averiguar las características de los dispositivos hace necesario un sistema de reconocimiento paralelo para poder estar seguros de que las posibilidades técnicas de los dispositivos son correctamente detectadas. Un ejemplo de un sistema basado en el fichero RDF es User Agent Profile (UAProf). Un fichero UAProf detalla las características de un dispositivo móvil. UAProf está relacionado con Composite Capabilities/Preferences Profile (CC/PP), que es una extensión de vocabulario del fichero RDF, enfocada en definir características y preferencias. En lugar de determinar las características basándose en el aparato, CC/PP involucra al dispositivo, el cual realmente cuenta al servidor cuáles son sus características. El gran problema de CC/PP es que no es ampliamente admitido por los dispositivos. Más específicamente, no todos los dispositivos disponen de UAProf's. Además, UAProf es un sistema que no es considerado lo suficientemente maduro ni estable como para que los sistemas de producción dependan de él. Por todas estas razones, esta forma de detectar las características de los dispositivos no ha sido la elegida para la realización de este proyecto.
- Mediante los agentes de usuario (user-agent): un user-agent es una aplicación cliente que implementa un protocolo de red usado en comunicaciones dentro de un sistema distribuido cliente-servidor. Cuando se solicita una página web, todos los navegadores

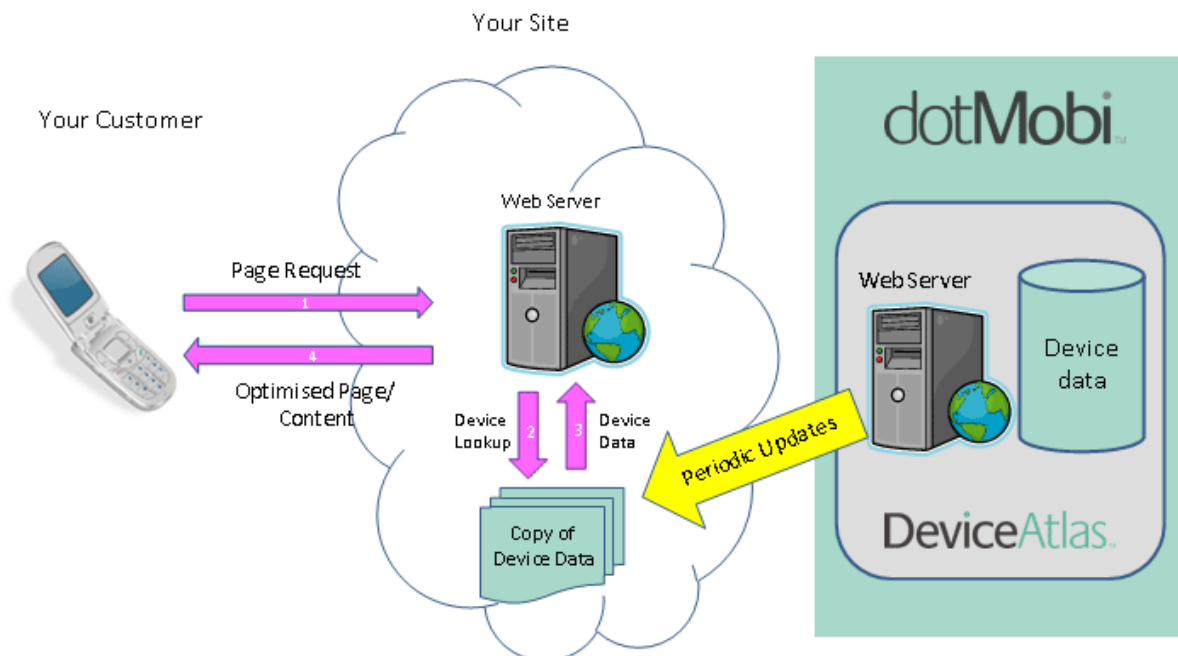
envían su “user-agent”, una cadena de texto que identifica al dispositivo desde el que se hace la petición, a los ojos del servidor. Dicha cadena de texto pertenece a la primera petición HTTP, y tiene la particularidad de que es única para cada dispositivo. Si se utiliza PHP, el agente de usuario se puede encontrar en una variable global llamada “\$\_SERVER[HTTP\_USER\_AGENT]”. Tras obtener la cadena de texto user-agent, teniendo acceso a una base de datos de todos los dispositivos móviles, junto a sus características, el servidor podría buscar dicha cadena de texto y conseguir la información relacionada con ese user-agent, con el fin de conocer todas las características del dispositivo que está accediendo al sitio web. El único inconveniente que se podría plantear es que la base de datos de móviles debería ser mantenida y actualizada, para que siguiera siendo precisa con el paso del tiempo. En cualquier caso, actualmente existen unas cuantas alternativas entre las que elegir cuando se habla de bases de datos de dispositivos móviles, algunas de las cuales son gratuitas y de código abierto.

#### **3.4.1.1 Bases de datos de dispositivos**

- DeviceAtlas: Es una base de datos de dispositivos comercial, y su correspondiente API para tratarla, que proclama ser la “primera base de datos de dispositivos móviles mundial”. La base de datos DeviceAtlas es una agregación de información de dispositivos proveniente de operadoras, fabricantes, WURFL (más adelante se explicará en que consiste la base de datos WURFL), y otras fuentes. DeviceAtlas suministra datos del dispositivo detectado, que van desde la resolución de la pantalla hasta la cadena de texto user-agent, mediante la creación de un fichero de texto plano JSON. Las características del dispositivo son almacenadas jerárquicamente, de forma similar a como lo hace WURFL, para asegurar un fichero de datos JSON conciso. DeviceAtlas contiene datos exhaustivos de más de 5000 dispositivos móviles actualmente en uso. Su API es proporcionada para ser soportada por los lenguajes de programación más comunes (Java, PHP, .Net). Existe una licencia gratuita de evaluación, disponible solo para propósitos de desarrollo y prueba, pero es necesario pagar una cuota anual si se requieren todos los servicios que la base de datos puede ofrecer, como por ejemplo actualizaciones.
- DetectRight: Su base de datos está almacenada en el servidor de DetectRight. Las características de los dispositivos son obtenidas tras hacer una petición de consulta a la base de datos, consiguiendo como resultado un fichero XML. DetectRight suministra

librerías para los lenguajes de programación PHP y .Net. En su sitio web proclaman tener más de 20000 dispositivos incluidos en su base de datos. Al igual que con DeviceAtlas, hay una licencia gratuita con características muy limitadas, teniendo que pagar para conseguir tener acceso a servicios extra.

- WURFL (Wireless Universal Resource FiLe): Es un proyecto de código abierto que provee un fichero de configuración XML que contiene información sobre las características de dispositivos móviles. WURFL contiene más de 500 datos distintos para cada dispositivo, y hay miles de dispositivos. Es completamente gratuito, y últimamente la base de datos está siendo actualizada mensualmente con los últimos dispositivos que la comunidad va incluyendo. Dispone de muchas APIs para su soporte en multitud de lenguajes de programación (Java, PHP, .Net, Perl, Ruby, Python, C++...). Además, aparte de las APIs estándar, existen librerías interesantes basadas en WUFL desarrolladas por la comunidad, de las cuales un ejemplo es Tera-WURFL. Dicha librería mejora los tiempos de acceso a la base de datos WURFL, comparados con el uso de la API “normal” para PHP, creando una base de datos MySQL con los datos extraídos del fichero XML.



**Figura 8: Diagrama de funcionamiento de una base de datos de terminales móviles**

En la figura 8 se puede apreciar el funcionamiento de la base de datos DeviceAtlas, pero es extensible al caso de WURFL, ya que funciona de forma idéntica: periódicamente se

ofrece una actualización de la base de datos, de la cual hay que mantener una copia local en el servidor para hacer las consultas cada vez que se reciba una petición de un dispositivo.

### **3.4.1.2 WURFL**

Por todas las razones expuestas anteriormente, WURFL es la base de datos elegida para obtener las características de los dispositivos que van a acceder al sitio web en este proyecto. Sus cualidades son que es gratuito, de código abierto, y que es bastante fiable, sobre todo para los elementos que van a estar involucrados en este proyecto. WURFL es casi una base de datos de código abierto estándar, y hay una gran comunidad detrás de ella, respaldándola y haciendo uso de ella. En este momento, mantener una base de datos no es uno de los objetivos de este proyecto, por ello, en lugar de trabajar con la librería Tera-WURFL, con la que hay que mantener la base de datos en MySQL, la API PHP de WURFL es la solución elegida y con la que se va a trabajar de ahora en adelante.

Los datos en el fichero WURFL son jerárquicos, lo que significa que los dispositivos están organizados en una estructura de árbol, siendo algunos de los nodos del árbol, familias de dispositivos. De esta forma, si alguno de los dispositivos no tiene una característica específica definida, el dispositivo simplemente hereda el valor de esa característica de su nodo padre. Por otra parte, para cada dispositivo, las características están clasificadas en grupos, pero esta clasificación solo pretende mejorar el entendimiento de la base de datos XML por parte de las personas, no tiene ninguna otra función.

Cualquiera podría modificar su copia privada de WURFL, pero entonces, cuando una nueva versión actualizada de la base de datos WURFL estuviera disponible para la comunidad, y hubiera que descargarla y ponerla en el lugar de la copia modificada anterior, las modificaciones aplicadas a la copia antigua de la base de datos se perderían. Para evitar esta situación WURFL ofrece la posibilidad de crear parches para la base de datos original con el objetivo de sobrescribir cualquier valor de la base de datos original, o incluso crear nuevos dispositivos, características, grupos, etc. Estos parches son simplemente ficheros XML que siguen la misma estructura que la base de datos WURFL. Cuando se crea la caché de la base de datos, para permitir accesos más rápidos, los datos de los parches también son importados para construir una versión modificada de la base de datos original, pero el proceso de creación de la caché será explicado en profundidad más adelante. WURFL generalmente engloba a dispositivos móviles. Sin embargo, para desarrolladores que quieran distinguir entre

un navegador de un dispositivo móvil y un navegador de escritorio, WURFL provee un parche para los navegadores web. Este parche se comporta como una extensión de WURFL que añade los agentes de usuario de navegadores web de escritorio más comunes.

### 3.4.1.3 Configuración de WURFL

Estos son los pasos seguidos para instalar, configurar y usar WURFL y su API para PHP:

1. Descargar la API para PHP a una carpeta accesible en el servidor Apache. La versión de la API descargada y usada en el proyecto fue la 1.1, y la descarga se realizó desde el sitio web SourceForge para el proyecto WURFL en <http://sourceforge.net/projects/wurfl/files/WURFL%20PHP/>
2. Descargar la última versión de la base de datos WURFL y el parche para los navegadores web a una carpeta accesible en el servidor Apache. Crear un directorio con permisos de escritura en el que se creará la caché de los datos de la base de datos WURFL.
3. Configurar la API personalizando el fichero de configuración WURFL, "wurfl-config.xml", en el que se determina la localización de los ficheros de la base de datos principal WURFL y de los parches, así como la ruta del directorio que almacenará la base de datos cacheada en el disco duro. El contenido del fichero de configuración usado en el proyecto es el que aparece a continuación:

```
<wurfl-config>
  <wurfl>
    <main-file>wurfl-latest.zip</main-file>
    <patches>
      <patch>web_browsers_patch.xml</patch>
    </patches>
  </wurfl>
  <persistence>
    <provider>file</provider>
    <params>dir=./cache</params>
  </persistence>
  <cache>
    <provider>file</provider>
    <params>dir=./cache</params>
  </cache>
</wurfl-config>
```

Este fichero de configuración indica que la base de datos WURFL principal, el fichero del parche y la memoria caché en disco, están todos localizados en el mismo directorio que el fichero de configuración. El elemento <main-file> especifica la localización de la base de datos WURFL principal. El elemento <patch> especifica la

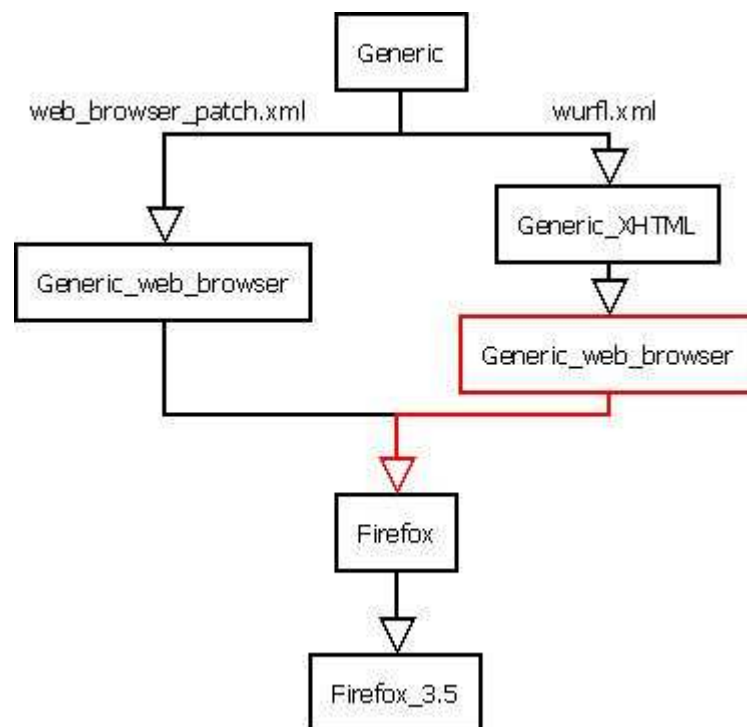
localización del parche usado para modificar la base de datos. El elemento `<persistence>` especifica el mecanismo de caché usado para almacenar los datos después de aplicar el parche, y en este caso, la carpeta donde los ficheros de la caché van a ser almacenados, ya que el sistema de caché seleccionado ha sido del tipo “file”. El elemento `<cache>` especifica la implementación caché utilizada para almacenar valores que han sido solicitados previamente a la base de datos para consultas más rápidas, y también la carpeta donde estos ficheros de caché van a ser almacenados, ya que el tipo de caché seleccionada ha sido “file”.

Es de destacar que, en el fichero de configuración, el fichero principal de la base de datos no es un fichero XML, sino un fichero Zip. El sitio web de WURFL recomienda que la instalación PHP tenga soporte para los ficheros Zip. Remarcan que no es un requisito, pero lo aconsejan para evitar tener que tratar el fichero XML de WURFL, que es de gran tamaño. Para la última versión de la base de datos el tamaño del fichero XML es de 14 Megabytes, mientras que la versión comprimida del mismo en Zip ocupa solo 357 Kilobytes, 40 veces menos.

Dado que WURFL trata acerca de los dispositivos móviles, desde su sitio web proveen un parche para los desarrolladores que quieran detectar si un user-agent pertenece a un terminal móvil o a un navegador web de escritorio (Mozilla Firefox, Internet Explorer...). El nombre del fichero que representa al parche es `web_browsers_patch.xml`, y es el que se especifica en el fichero de configuración de WURFL. Durante el desarrollo de este proyecto, ha sido necesario modificar el fichero original `web_browsers_patch.xml` porque se detectó un error cuando estaba siendo probado si los navegadores web eran correctamente reconocidos usando la base de datos WURFL. Estas pruebas mostraron que WURFL era certero cuando se trataba de conseguir las características de un dispositivo móvil, pero desde un navegador web de escritorio algunas de las características aparecían con múltiples valores, en lugar de con un solo valor, como era de esperar.

Después de estudiar la estructura de los ficheros de la base de datos, `wurfl.xml` y `web_browsers_patch.xml`, se pudo detectar qué había causado el extraño resultado. El problema era que había una clase con el mismo nombre en ambos ficheros, pero con diferentes valores para una misma característica: “Generic\_web\_browser” era el nombre de la clase problemática. Así, las clases que heredaban de la clase

"Generic\_web\_browser" estaban obteniendo dos valores para características que se supone que tienen un solo valor. Por ejemplo, como se puede ver en el diagrama de la figura 9, la clase "Firefox" heredaba de "Generic\_web\_browser", en wurfl.xml, una resolución de pantalla de 770 x 300, pero de acuerdo a la clase "Generic\_web\_browser" en web\_browsers\_patch.xml, heredaba una resolución de 800 x 600. El resultado era un array de dos elementos en lugar de un elemento simple. Incluso algunas características booleanas eran incorrectas, como por ejemplo la característica "is a mobile device" era un array de dos elementos, siendo ambos elementos el valor "false", por lo que no era posible usar esta característica como un booleano directamente, como se hacía en el código, sino que debería ser tratada como un array de booleanos.

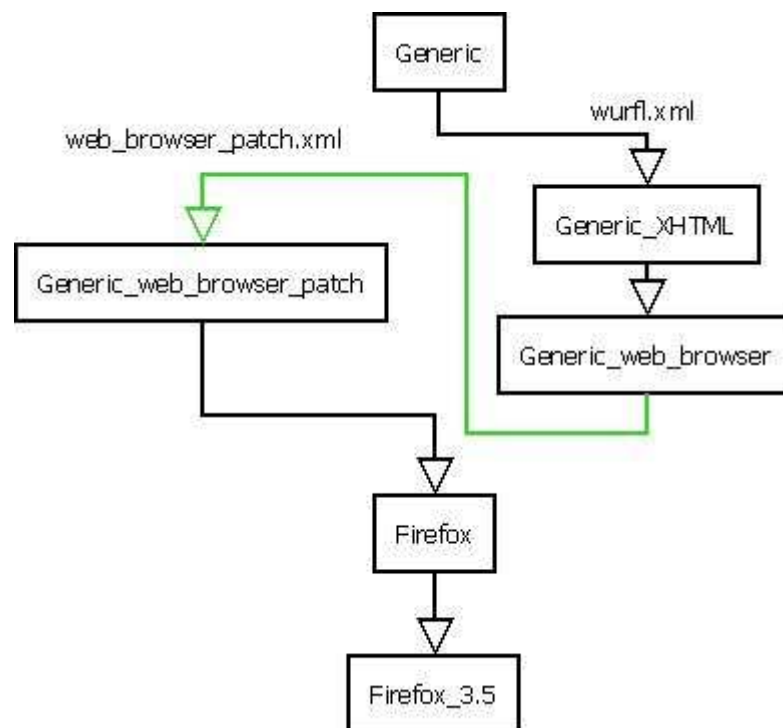


**Figura 9: Diagrama de la estructura original de la base de datos WURFL**

Para resolver estos problemas, fue aplicado un cambio a la estructura de las clases. El objetivo era que se heredara la información de las características desde la clase "Generic\_web\_browser" perteneciente al parche web\_browsers\_patch.xml, sobrescribiendo la información heredada desde la clase "Generic\_web\_browser" del fichero wurfl.xml, ya que eso es lo que se pretende siempre con la creación de un parche, que se sobrescriban los valores del original. Entonces, para conseguir esto, el nombre de la clase duplicada en el parche ha sido modificado a



"Generic\_web\_browser\_patch", y su nodo padre ha sido cambiado también, siendo ahora su padre la propia clase "Generic\_web\_browser", ya no duplicada, perteneciente a la base de datos original wurfl.xml, para permitir que se hereden valores de la base de datos original pero que se de preferencia a los posibles valores distintos que en el parche se hayan definido, como se muestra en el diagrama de la figura 10. Aparte de esto, obviamente fue necesario actualizar todas las clases en el fichero del parche cuyo nodo padre era "Generic\_web\_browser", al nuevo nombre de su clase padre, "Generic\_web\_browser\_patch".



**Figura 10: Diagrama de la estructura de la base de datos tras aplicar la corrección**

De esta forma, el fichero wurfl.xml puede ser actualizado sin temor, ya que esta nueva estructura no va a ser cambiada, puesto que la clave está en el parche de los navegadores web. Si apareciera un nuevo parche, con nuevas versiones de navegadores web, la estructura del mismo debería ser analizada para ver si sería necesario aplicar las mismas modificaciones.

4. Escribir código PHP que inicialice la API de WURFL usando el fichero wurfl-config.xml (o su lugar, ejecutando un fichero por defecto que WURFL suministra para esta tarea, DefaultWURFLReloader.PHP). El fichero PHP que se ha usado durante la realización de este proyecto ha sido el siguiente:

```

<?php
require_once "wurfl-php-1.r1/WURFL/WURFLManagerProvider.php";
  
```

```
$wurflConfigFile = "/home/javier/public_html/wurfl-php-1.r1/examples/resources/wurfl-  
config.xml"; //provide the absolute or relative path to the config file.  
$wurflManager = WURFL_WURFLManagerProvider::getWURFLManager($wurflConfigFile);  
?>
```

Nótese que el directorio de instalación de la API de WURFL en la sentencia `require_once` es relativo a la localización del fichero `wurfl-config.xml` que contiene este código PHP (aunque se podría especificar una ruta absoluta; son igualmente válidas en PHP). La localización del fichero de configuración es una ruta absoluta.

5. Escribir código PHP que reconozca un dispositivo móvil y obtenga sus características. A continuación se añade un ejemplo en PHP, pero antes se incluye una lista de diferentes métodos que la API de WURFL provee para el reconocimiento de dispositivos:

La primera variante utiliza directamente a la cadena de texto `user-agent` para identificar al dispositivo, como en el siguiente código PHP:

```
$requestingDevice = $wurflManager->getDeviceForUserAgent  
($_SERVER['HTTP_USER_AGENT']);
```

Una forma similar, pero más simple, es usar las cabeceras de la primera petición HTTP enviadas desde el dispositivo al servidor para identificar el dispositivo, permitiéndole buscar el `user-agent` por sí mismo, como en este ejemplo en PHP:

```
$requestingDevice = $wurflManager->getDeviceForHttpRequest($_SERVER);
```

Hay una tercera forma para el reconocimiento de dispositivos, solo válida cuando el identificador WURFL es conocido. En el ejemplo que sigue, se ha usado el identificador del Nokia N810:

```
$requestingDevice = $wurflManager->getDevice("nokia_n810_ver1_subua2");
```

En todos los casos, la variable `$requestingDevice` es un objeto `WURFL_Device` que puede ser usado para obtener las características del dispositivo, llamando al método `getCapability`. Dicho método espera el nombre de una característica como parámetro, y devuelve el valor de la característica solicitada.

El siguiente código PHP ha sido el utilizado en el proyecto para obtener algunas características interesantes, y también para detectar si la petición estaba siendo hecha desde un dispositivo móvil o desde un ordenador:

```

<?php
$requestingDevice = $wurflManager->getDeviceForHttpRequest($_SERVER);
?>

<div id="content">
Requesting Browser User Agent: <b> <?php echo $_SERVER["HTTP_USER_AGENT"] ?>
</b>
<ul>
<li>ID: <?php echo $requestingDevice->id ?> </li>
<li>Brand Name: <?php echo $requestingDevice->getCapability("brand_name") ?> </li>
<li>Model Name: <?php echo $requestingDevice->getCapability("model_name") ?> </li>
<li>Xhtml Preferred Markup: <?php echo $requestingDevice->
getCapability("preferred_markup") ?> </li>
<li>Resolution Width: <?php echo $requestingDevice->getCapability("resolution_width") ?>
</li>
<li>Resolution Height: <?php echo $requestingDevice->getCapability("resolution_height") ?>
</li>
<li>Pointing Method: <?php echo $requestingDevice->getCapability("pointing_method") ?>
</li>
<li>Support Javascript: <?php echo $requestingDevice->
getCapability("ajax_support_javascript") ?> </li>
<li>Support Cookies: <?php echo $requestingDevice->getCapability("cookie_support") ?>
</li>
<li>Dual Orientation: <?php echo $requestingDevice->getCapability("dual_orientation") ?>
</li>
</ul>

<?php
$is_wireless = $requestingDevice->getCapability("is_wireless_device");
$claims_web_support = $requestingDevice->getCapability("device_claims_web_support");
if (($is_wireless == "false") && ($claims_web_support == "true"))
    echo "The request is from a desktop PC";
else
    echo "The request is from a mobile device";
?>
</div>

```

Accediendo a este ejemplo en PHP desde un dispositivo es posible obtener como resultado su user-agent, su marca y modelo, su resolución de pantalla, su forma de moverse por la pantalla, si admite JavaScript y cookies o no, y mas características técnicas interesantes. Después, se comprueba si la petición viene de un dispositivo móvil o no. Para llevar a cabo esta tarea, de acuerdo al sitio web de WURFL, hay que tener en cuenta dos características: "is\_wireless\_device" y "device\_claims\_web\_support". A continuación se incluyen algunos ejemplos de la ejecución de este script:

Primero, en la figura 11, desde un Nokia N810.

Requesting Browser User Agent: **Mozilla/5.0 (X11; U; Linux i686; en-GB; rv:1.9.1.8) Gecko/20100214 Ubuntu/9.10 (karmic) Firefox/3.5.8**

- ID: nokia\_n810\_ver1\_subua2
- Brand Name: Nokia
- Model Name: N810
- Xhtml Preferred Markup: html\_wi\_w3\_xhtmlbasic
- Resolution Width: 800
- Resolution Height: 480
- Pointing Method: stylus
- Support Javascript: true
- Support Cookies: true
- Dual Orientation: true

The request is from a mobile device

Figura 11: Ejemplo de detección del dispositivo desde un Nokia N810

Y después, en la figura 12, desde un portátil con navegador Mozilla Firefox 3.5.8.

Requesting Browser User Agent: **Mozilla/5.0 (X11; U; Linux i686; en-GB; rv:1.9.1.8) Gecko/20100214 Ubuntu/9.10 (karmic) Firefox/3.5.8**

- ID: firefox\_3\_5
- Brand Name: firefox
- Model Name: 3.5
- Xhtml Preferred Markup: html\_web\_4\_0
- Resolution Width: 800
- Resolution Height: 600
- Pointing Method: mouse
- Support Javascript: true
- Support Cookies: true
- Dual Orientation: false

The request is from a desktop PC

Figura 12: Ejemplo de detección del dispositivo desde el navegador Firefox 3.5.8

Después de probar con varios dispositivos más, se puede concluir que la base de datos WURFL es bastante exacta. Detecta el dispositivo perfectamente obteniendo la marca y modelo del mismo. Hay algunas características que no son completamente correctas en algunos casos concretos, por ejemplo, el Nokia N80 y el Nokia E61 admiten cookies pero, en las pruebas realizadas, la base de datos WURFL asegura que no. No obstante, las características más importantes, como la resolución de pantalla, son acertadas, así como a la hora de distinguir si la petición ha sido hecha desde un dispositivo móvil o desde un navegador web de escritorio.

#### 3.4.1.4 Rendimiento de la base de datos WURFL

Una vez configurada la base de datos WURFL se han realizado una serie de pruebas para determinar el rendimiento de la misma. Estas pruebas han consistido en la ejecución de un script PHP que medía el tiempo que se tardaba en acceder a la base de datos. Dicho script se adjunta en el apéndice C de esta memoria. Se ha repetido esta prueba para distintos agentes de usuario, y para ello se ha utilizado el plugin de Firefox “user-agent switcher”, así se ha podido comprobar la variabilidad del tiempo empleado en las búsquedas, dependiendo del agente de usuario que se esté buscando.

Para cada agente de usuario, se han realizado 3 pruebas distintas. Debido a la velocidad de una búsqueda, es preferible realizar varias consultas a la vez. De esta forma, en el primero de los casos se han realizado 100 consultas, para hacer después 500, y al final 1000 consultas. Además, para cada uno de los 3 casos, en el script, se repite la prueba 3 veces, siendo el resultado final la media de las 3 pruebas. Por ejemplo, para el tiempo que se tarda en realizar 100 consultas, se mide 3 veces el tiempo al hacer 100 consultas, calculando la media de los 3 datos. De esta forma se consigue eliminar datos extremos que no correspondan con la realidad.

Otro dato que es conveniente conocer, antes de dar los resultados de las pruebas de rendimiento de la base de datos, es el referente a las características técnicas del servidor en el que se realizan dichas pruebas. El servidor es simplemente un ordenador portátil, ofrecido por la empresa para la realización de este proyecto. A pesar de no conocer con exactitud las especificaciones del equipo, el ordenador portátil no es ni mucho menos un último modelo, sino un equipo de hace varios años, por lo que hay que tener esto en cuenta a la hora de valorar los resultados.

En la tabla 1 se pueden observar los resultados de las primeras pruebas realizadas con la base datos WURFL, en este caso sin utilizar el sistema de caché que ofrece. En dicha tabla aparece el tiempo medido en segundos.

Dispositivo	100 consultas	500 consultas	1000 consultas
Mozilla 3.5.9	1,2288	5,7514	11,5285
Internet Explorer 8	1,1693	5,8937	11,6826

Iphone 3.0	1,1199	5,5936	11,2394
Nokia e61	1,3514	6,6533	13,2627
Nokia n80	1,4166	7,1661	14,6590
Nokia 810	1,6568	8,4764	16,9375
MDA Vario	0,9838	4,8287	9,7058
MDA III	1,1525	5,8405	11,6836
<b>Tiempo medio</b>	<b>1,25988</b>	<b>6,27547</b>	<b>12,58739</b>

Tabla 1: Rendimiento de la base de datos WURFL sin caché

Se puede observar que, en realizar 100 consultas, se tarda entre 0,98 y 1,66 segundos. El tiempo medio para 100 consultas es de 1,26 segundos, por lo que puede deducirse que el tiempo que se tarda en realizar una consulta es de 0,0126 segundos ( $1,26 / 100 = 0,0126$ ). El resto de mediciones obtenidas en la tabla concuerdan con este dato, teniendo en cuenta el redondeo, ya que para el caso de 500 consultas,  $0,0126 * 500 = 6,3$  segundos, y para el caso de 1000 consultas,  $0,0126 * 1000 = 12,6$  segundos.

Cabe destacar que los resultados son independientes del user-agent del dispositivo, e incluso para el caso en el que el user-agent pertenece a un navegador de ordenador, como son Mozilla 3.5.9 e Internet Explorer 8, el tiempo empleado no es necesariamente menor, ni mayor, comparado con el de cualquier dispositivo. Hay que recordar que estos navegadores son incluidos en la base de datos a través de un parche, circunstancia que, a la vista de los resultados, no implica nada en cuestión de rendimiento.

Estas pruebas se repitieron para probar el sistema de cache que ofrece WURFL. La tabla 2 muestra los nuevos resultados.

Dispositivo	100 consultas	500 consultas	1000 consultas
Mozilla 3.5.9	1,1060	5,6978	10,8910
Internet Explorer 8	1,0959	5,4964	10,9703
Iphone 3.0	1,0822	5,2872	10,3459
Nokia e61	0,9703	4,6843	9,5523
Nokia n80	0,8235	3,9925	7,8472
Nokia 810	1,0061	5,1199	10,1600
MDA Vario	0,7843	3,9097	7,8533
MDA III	0,9728	4,9072	9,7644
<b>Tiempo medio</b>	<b>0,98013</b>	<b>4,88687</b>	<b>9,67305</b>

Tabla 2: Rendimiento de la base de datos WURFL con caché

En este caso se puede apreciar que todos los tiempos de consulta han disminuido en mayor o menor medida. Así, se observa que, para realizar 100 consultas, el tiempo mínimo necesario ha sido 0,78 segundos, mientras que el máximo ha sido de 1,1 segundos. La media del tiempo necesario para 100 consultas ha sido ahora de menos de 1 segundo, 0,98 segundos exactamente. De esta forma se puede calcular que el tiempo medio para 1 consulta es 0,0098 segundos ( $0,98 / 100 = 0,0098$ ). Al igual que antes, este dato concuerda con los obtenidos para 500 y 1000 consultas, teniendo en cuenta los redondeos y las mínimas variaciones de los datos:  $0,0098 * 500 = 4,9$  segundos, y  $0,0098 * 1000 = 9,8$  segundos.

A continuación se ha obtenido la diferencia de tiempo obtenida tras el uso de la caché de WURFL. Para analizar la mejora que supone su uso, se incluye la tabla 3 con los segundos de mejora para todos los casos probados.

Dispositivo	100 consultas	500 consultas	1000 consultas
Mozilla 3.5.9	-0,12280	-0,05359	-0,63745
Internet Explorer 8	-0,07345	-0,39731	-0,71228
Iphone 3.0	-0,03768	-0,30640	-0,89349
Nokia e61	-0,38105	-1,96899	-3,71039
Nokia n80	-0,59313	-3,17356	-6,81182
Nokia 810	-0,65074	-3,35657	-6,77754
MDA Vario	-0,19950	-0,91908	-1,85248
MDA III	-0,17967	-0,93331	-1,91925
<b>Tiempo medio</b>	<b>-0,27975</b>	<b>-1,38860</b>	<b>-2,91434</b>

**Tabla 3: Mejora en el rendimiento de la base de datos WURFL al utilizar la caché**

Se puede observar en la tabla que la mejora es perceptible en todos los casos, por el hecho de que todos los resultados son negativos, ya que la operación realizada ha sido “tiempo sin sistema cache” – “tiempo con sistema caché”. Para 100 consultas se tienen mejoras de entre 0,04 y 0,6 segundos, lo que da una idea de la amplitud del rango de mejora que puede experimentar un dispositivo en concreto, que es algo que no se puede saber con antelación. El tiempo medio de mejora para 100 consultas es de 0,28 segundos, lo que implica reducir el tiempo de las consultas en un 22%.

Un hecho que sí se puede advertir en esta tabla es que la reducción de los tiempos de consulta es aparentemente lineal, pero en determinados dispositivos concretos se pueden apreciar ciertas incongruencias. Estas incongruencias pueden ser debidas a la forma en que

WURFL usa la memoria caché, hecho que queda fuera del alcance de este proyecto. No obstante, siguiendo la media de los tiempos de mejora de todos los dispositivos, sí es apreciable la linealidad de la mejora.

Para intentar actualizar estos resultados, se intentó realizar las mismas pruebas en el ordenador del tutor del proyecto en la compañía. Sin embargo, los resultados obtenidos fueron peores, y es que, a pesar de ser un portátil potente y moderno, durante las consultas solo era utilizado uno de los cuatro núcleos de los que constaba el procesador del equipo. Por esta razón se desestimó el realizar pruebas en los servidores del departamento en el que se realizaba el proyecto, ya que estos tenían también procesadores de cuatro núcleos.

Por otra parte, el tutor del proyecto en la compañía realizó las mismas pruebas, pero utilizando la librería de la base de datos WURFL en Java, y los resultados obtenidos fueron mejores, del orden de 500 consultas por segundo. Presumiblemente, Java incorpora un sistema de caché propio que mejora los tiempos de consulta, además de aprovechar los cuatro núcleos del procesador del equipo, cosa que PHP no es capaz de hacer.

Una vez que se tiene configurada y probada la base de datos WURFL, que se usará para detectar las características de los dispositivos que accedan a la página, el siguiente paso en la solución planteada es clasificar todo el espectro de dispositivos en categorías de dispositivos con similares características.

### **3.4.2 Identificación de las categorías de dispositivos móviles**

Para poder definir las categorías en las que enmarcar a los distintos dispositivos móviles se debe estudiar previamente las características técnicas que definirán a cada categoría, o dicho de otro modo, las propiedades más significativas que los dispositivos móviles pueden tener o no, y que afectan a la hora de navegar por Internet.

#### **3.4.2.1 Análisis de las características más importantes de los dispositivos**

A continuación se listan las características técnicas principales de los móviles a tener en cuenta:

- Resolución de pantalla: Posiblemente la característica más importante, puesto que de ella depende el que una página web se ajuste a la pantalla del dispositivo, o que haya



que hacer scrolling usando las barras de desplazamiento, siendo especialmente molesto el scrolling horizontal, ya los usuarios no están acostumbrados a él tanto como al vertical, el cual es más aceptado debido a su frecuente aparición tanto en páginas web, como en documentos de cualquier tipo, cuando se usan ordenadores de sobremesa y portátiles. Así pues, la experiencia de usuario depende en gran medida del acierto a la hora de adecuar la página a la anchura y longitud de la pantalla del dispositivo que se esté utilizando.

- JavaScript, Flash: Las tecnologías que se utilizan en Internet pueden ser perfectamente soportadas por los navegadores para ordenadores, pero puede suceder que, en el caso de los navegadores web para dispositivos móviles, no sean correctamente soportadas, obteniendo como resultado un visionado incorrecto de algunas zonas de las páginas web en dichos dispositivos, o incluso la incapacidad de visualizar la página directamente. Los sitios web que no utilizan JavaScript ni Flash son, en apariencia, más básicos o más simples, ya que su uso permite mostrar animaciones o, en general, funcionalidades añadidas a las ofrecidas por el lenguaje HTML. Por eso, estas tecnologías web son un añadido para la calidad final de una página web, pero no son imprescindibles en absoluto, y la experiencia de usuario puede ser buena incluso en el caso de no hacer uso de ellas.
- Orientación de la pantalla: Los últimos modelos de teléfonos móviles, aquellos con pantalla más grande, suelen permitir voltear el terminal y que la orientación de la pantalla se ajuste automáticamente a la orientación física del aparato gracias a un acelerómetro interno que incorporan. Esto puede suponer una dificultad, ya que la experiencia de usuario varía al visitar una página web desde una pantalla en la que el ancho es más grande que el alto, comparándola con el caso contrario, en el que la altura de la pantalla sea más grande que la anchura. En general, es el propio usuario el que decide la orientación para que el visionado de la web sea óptimo, pero debería ser al revés, que la web se adaptase a la orientación con la que quiere usar el dispositivo el usuario. Por todo esto, la orientación debería poder detectarse, aunque como se discutirá posteriormente, esto entraña cierta dificultad.
- Forma de introducción de texto: Originalmente, los teclados de los móviles constan de una tecla por cada número, y no una tecla por cada letra como en los teclados de los ordenadores, con lo que se agrupan 3 ó 4 caracteres en cada tecla. Algunos móviles incorporan un teclado QWERTY completo, como por ejemplo las típicas Blackberries, u otros móviles que disponen de un teclado deslizante debajo de la pantalla. Incluso

otros dispositivos, como por ejemplo los smartphones con pantalla táctil, prescinden del teclado físico, ofreciendo un teclado virtual. Así, dependiendo de la forma de introducir texto, la sensación al escribir es completamente distinta. No obstante, esta característica no afecta tanto a la navegación por la red propiamente dicha, y lo que es más importante, poco se puede hacer desde el punto de vista de la usabilidad de una página web, más que tener en cuenta, a la hora del diseño del sitio web, que esa página va a ser accedida desde dispositivos con distintas maneras de introducir texto, para evitar en la medida de lo posible el crear un diseño excesivamente cerrado, y que excluya, o cree problemas, a ciertos usuarios potenciales.

- **Método de navegación:** En función del método de navegación del dispositivo, la forma de controlar el dispositivo varía, y con ello la forma de navegar por Internet con el mismo. El método más básico podría ser mediante botones que permitan el movimiento en las cuatro direcciones principales: arriba, abajo, derecha e izquierda. A partir de aquí, los fabricantes idean una variedad de formas para navegar en los dispositivos, como son mediante un joystick o pequeña palanca, mediante una rueda al estilo Blackberry, o a través de la pantalla táctil, ya sea pulsándola con los dedos directamente, o con un estilete (o stylus). Cabe reseñar que el método de navegación tiene una fuerte implicación con la forma de interactuar con el dispositivo, y con la variedad de acciones que pueden ser ejecutadas: un usuario de ordenador, con un ratón, puede realizar acciones que no se pueden llevar a cabo de la misma forma con los métodos de navegación de los móviles, como por ejemplo arrastrar objetos; e incluso hay diferencias entre los móviles, ya que la sensación al mover un puntero por la pantalla con un joystick es totalmente distinta a la de manejar un móvil con pantalla táctil, en la que ni siquiera existe el puntero, ya que no es necesario. Otro punto a destacar es la necesidad de adecuar los diseños de las páginas web orientadas a los móviles al método de navegación en los mismos, puesto que no se tiene la misma precisión para pulsar un botón en la pantalla con un stylus, que con un dedo, con lo que, por ejemplo, en un iPhone los botones y los enlaces no deberían ser excesivamente pequeños. Esto es sin tener en cuenta lo ajustado que sea el sistema táctil, porque aunque sea muy ajustado y esté muy bien calibrado, el usuario podría tener ciertas dudas del lugar exacto en el que pulsar a la hora de seguir un enlace que es demasiado pequeño, y esto es algo que se debe evitar.

### **3.4.2.2 Categorías de dispositivos y sus características**

Una vez analizadas las características que se pueden tener en cuenta a la hora de definir las diferentes categorías, hay que decidir las particularidades de cada categoría. Durante el análisis de las características se ha comprobado que, tanto la orientación como la resolución de la pantalla, son aspectos fundamentales para la creación de las distintas categorías, puesto que la futura página web adaptada debería aprovechar la mayor amplitud que ofrecen los dispositivos móviles con orientación paisaje (horizontal), esto es, cuya anchura es superior a la altura. En este tipo de dispositivos, la experiencia de usuario debería ser más cercana a la experimentada en un ordenador de sobremesa o portátil, ya que la relación de aspecto de estos es típicamente 16:9, o como poco 4:3, con lo que se asemeja bastante a la de los dispositivos con orientación panorámica, mientras que para los móviles que solo disponen de orientación retrato (vertical) la relación de aspecto puede ser aproximadamente la inversa.

Como una primera aproximación se decide utilizar la propia orientación principal del dispositivo móvil como característica determinante para decidir la categoría a la que pertenece. De esta forma, se clasifican los dispositivos en dos categorías, retrato y paisaje, de acuerdo a la orientación de la pantalla. Por otra parte, se hace necesaria la inclusión de una tercera categoría, una categoría por defecto, en la que quedarían enmarcadas todas las peticiones que no fueran hechas desde dispositivos móviles, es decir, para los casos en los que los usuarios estén utilizando un ordenador de sobremesa o un portátil, en cuyo caso no hay que adaptar la página original, ni realizar ninguna acción especial.

Una vez tomada esta decisión, se ha advertido de un detalle, y es que puede darse el caso de que algún dispositivo que pertenezca a la categoría de retrato, tenga una resolución de pantalla mayor en cuanto a anchura que otro que pertenezca a la categoría de paisaje, con lo que el primero sería más capaz de presentar la página web adaptada para la categoría de paisaje que el segundo. O podría suceder el caso contrario, en el que un dispositivo que pertenezca a la categoría de paisaje tenga una resolución de pantalla menor en cuanto a su anchura que los que pertenecen a la categoría de retrato, teniendo que la página adaptada para la categoría de paisaje se quedaría un poco grande para el primer dispositivo. Para que la adaptación de la página web sea óptima para el mayor número de dispositivos posibles, y teniendo en cuenta estos casos explicados, se ha decidido definir cuál sería la anchura mínima de pantalla que un dispositivo debería tener para mostrar correctamente la versión adaptada

para la categoría de paisaje, así como se ha definido también una anchura mínima de la página adaptada para la categoría de retrato, puesto que optar por una página más estrecha haría demasiado incómoda su visualización en los terminales más pequeños.

Para definir las anchuras mínimas de las dos categorías se utilizaron las primeras versiones de las páginas adaptadas, todavía no automáticamente, para cada categoría. Estas versiones serán explicadas en profundidad posteriormente, pero fueron modificaciones de la página original, a modo de bocetos, utilizando el editor HTML KompoZer manualmente. Con este editor, se ajustaron los elementos de la página para la categoría de paisaje al máximo para que fueran todavía reconocibles, y se comprobó que el tamaño mínimo idóneo para que el resultado fuera todavía elegante era de 500 píxeles de anchura. Para el caso de la página adaptada para la categoría de retrato, se redujo la anchura de los elementos igualmente al máximo, y se observó que la anchura mínima aceptable, para que la página resultante no perdiera identidad mostrando sus elementos excesivamente estrechos, era de 200 píxeles.

<b>Categoría</b>	<b>Características de la categoría</b>
<b>PC</b>	<i>Resolución de 1024 x 768. Página original, sin modificaciones.</i>
<b>Paisaje</b>	<i>Para móviles con orientación paisaje. Para móviles con orientación retrato cuya anchura sea mayor que 500.</i>
<b>Retrato</b>	<i>Para móviles con orientación retrato. Para móviles con orientación paisaje cuya anchura sea menor que 500. Anchura mínima de la página es 200.</i>

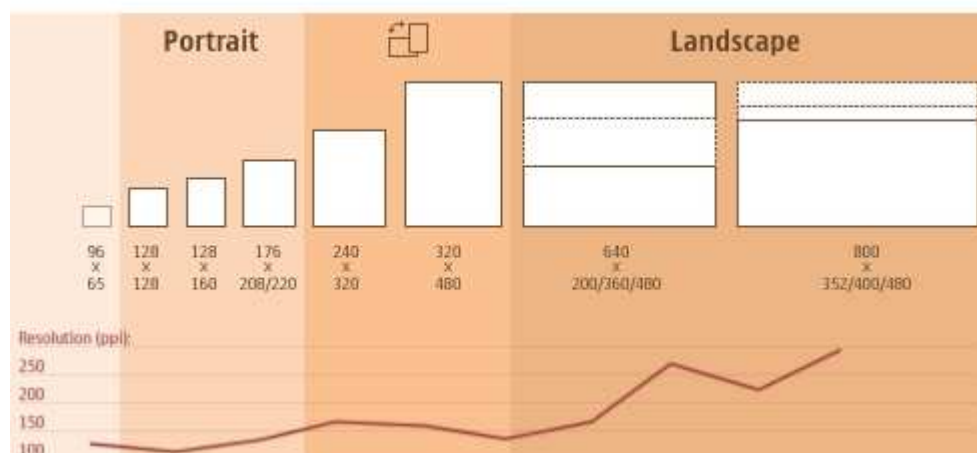
**Tabla 4: Características de las categorías**

No obstante, estas anchuras mínimas de las categorías implican cosas distintas. Mientras que la anchura mínima de 500 píxeles de la categoría de paisaje determina el límite inferior a partir del cual, incluso los dispositivos cuya orientación es paisaje, van a ser considerados en la práctica como si se tratara de dispositivos con orientación de retrato, el límite inferior de la categoría de retrato de 200 píxeles indica que la página adaptada para esta categoría no va a ser más estrecha que 200 píxeles. Esto implica que, en el caso de los dispositivos con anchura menor de 200 píxeles, la página que recibirán tendrá una anchura de 200 píxeles, por lo que no les aparecerá la totalidad de la anchura de la página web en pantalla, sino que les saldrá la barra de desplazamiento horizontal del navegador, y deberán desplazarse hacia la derecha para ver esa parte del sitio web.

Esta solución es la menos mala, ya que hay que tener en cuenta que los usuarios a los que se dará servicio con este proyecto tienen dispositivos con conexión Wi-Fi, con lo que se presupone una calidad mínima en los aparatos, es decir, es de esperar una cantidad muy pequeña de dispositivos con anchura de pantalla menor a 200 píxeles para el caso que nos ocupa. Por otra parte, el diseñar la página web con una anchura menor de 200 píxeles implicaría un cierto desorden en los elementos de la página, los cuales serían excesivamente estrechos, resultando en una página web sin claridad.

### 3.4.2.3 Tendencia de la resolución de pantalla de los dispositivos móviles

El gráfico de la figura 13 muestra la tendencia en cuanto a resoluciones de las pantallas de móviles a fecha de abril del año 2008. De acuerdo a la página web en la que se detalla el estudio, éste consta de alrededor de 400 modelos de móviles diferentes vendidos desde el año 2005.



**Figura 13: Resoluciones de pantalla típicas de los dispositivos móviles**

Al comentar el gráfico hay que tener en cuenta que, en el caso que nos ocupa, solo resultarían interesantes los dispositivos con conexión Wi-Fi. Por otra parte, es de esperar que los móviles con resoluciones de pantalla más bajas tengan, en general, peores características técnicas, por lo que, probablemente, la cantidad de ellos con capacidad para conectarse a redes Wi-Fi sea prácticamente nula.

Como se ve en el gráfico, en el estudio que han llevado a cabo, han diferenciado a los móviles por sus pantallas, observando las clases retrato, paisaje, y orientación dual. Debido a

que el gráfico del estudio fue realizado en 2008, 2 años antes de la realización de este proyecto, y a lo anteriormente mencionado respecto a que los móviles de peores características no tienen conexión Wi-Fi, se puede considerar la clase retrato de la gráfica como obsoleta para este proyecto. De esta forma, la clase orientación dual representaría la categoría retrato de este proyecto, con una resolución de pantalla mínima de 240 píxeles de ancho, ampliando un poco el margen en la categoría del proyecto, hasta 200 píxeles. No obstante, y a pesar de considerar obsoletos al resto de móviles con inferiores resoluciones, como se ha explicado anteriormente, todos ellos tendrían cabida en la categoría retrato, aunque la anchura de la página adaptada para ellos nunca será menor que 200 píxeles. La clase paisaje del gráfico concuerda con la categoría paisaje de este proyecto, en la que se enmarcan los dispositivos con una resolución de pantalla de más de 500 píxeles de anchura, que como se aprecia en la gráfica son típicamente 640 u 800 píxeles, con diversos valores para la altura.

En el estudio se apunta a que, en el momento en que se realizó, la resolución dominante en el segmento medio es de 240 x 320. Hoy en día se podría decir que dicha resolución sigue teniendo una posición dominante, pero no ya en solitario, sino junto a la resolución de pantalla de 320 x 480, que es la del iPhone 3GS. En poco tiempo las resoluciones “estándar” irán aumentando, puesto que los smartphones actuales ya incorporan resoluciones de pantalla superiores, así como la siguiente generación del iPhone, el iPhone 4, que ofrece una resolución de 640 x 960.

Esta gráfica, y el estudio que representa, ha sido de utilidad para tener una idea de cómo son las pantallas de los dispositivos móviles, pero en ningún momento durante la realización del proyecto se ha pretendido hacer un estudio exhaustivo de todo el mercado, ni siquiera obtener porcentajes de los dispositivos que caerían en una u otra categoría, datos que vendría muy bien conocer, pero que sería extremadamente complicado obtenerlos debido a la infinidad de distintos modelos existentes, y al desconocimiento de los usuarios reales de los puntos de acceso que interactuarán en el futuro con el sistema que se pretende desarrollar, así como por la rápida evolución de los móviles y sus características, que hace que cualquier estudio quede obsoleto al poco tiempo.

A pesar de esto, se ha creído oportuno comprobar qué dice la base de datos de dispositivos que se usa en el proyecto, WURFL, en cuanto a la variabilidad de las pantallas.

#### 3.4.2.4 Resolución de pantalla de los dispositivos de la base de datos WURFL

Se ha comprobado que, con un par de excepciones, como por ejemplo en el caso del Apple iPad, la resolución máxima de los dispositivos es de 800 píxeles de ancho, por lo que no se ha creído necesario el suponer que los dispositivos con una resolución cercana a la de los ordenadores de sobremesa, 1024 x 768, sean catalogados como si realmente fueran un PC, pertenecientes a dicha categoría, y así no realizar ninguna adaptación, sino mostrar directamente la página original, ya que sus características técnicas lo permiten. No obstante, debido a la rápida evolución de los dispositivos móviles, este sería un punto interesante a tratar en el futuro, como una posible mejora del sistema, y así se expone en el capítulo Conclusiones y trabajos futuros de esta memoria.

Modelo	Resolución de pantalla (ancho x alto, en píxeles)	Categoría
MDA III	240 x 320	Retrato.
MDA Vario	240 x 320	Retrato.
Nokia E61	320 x 240	Retrato.
Nokia N80	352 x 416	Retrato.
Nokia N810	800 x 480	Paisaje.
iPhone 3GS	320 x 480	Retrato. Caso especial.

**Tabla 5: Resolución de pantalla de los dispositivos utilizados en el proyecto**

En la tabla 5 se muestra un listado de los dispositivos que fueron suministrados por la compañía, y que han podido ser utilizados para hacer pruebas durante la realización del proyecto. Se incluye en dicha tabla la resolución de pantalla de los distintos dispositivos, obtenida a través de consultas a la base de datos WURFL, así como la categoría a la que pertenecen. El caso del iPhone 3GS es especial porque, a pesar de pertenecer a la categoría de retrato, hay que hacerle un ajuste específico cuando se adapta la página, debido a las particularidades de este dispositivo, pero esto se explicará más adelante, cuando se detalle el proceso de adaptación propiamente dicho. Hay que destacar también que, además de estos dispositivos, se ha podido utilizar un emulador de dispositivos Nokia, llamado Remote Device Access, que la propia marca ofrece para que los desarrolladores de aplicaciones hagan pruebas con sus modelos. Igualmente se ha utilizado un complemento del Firefox llamado

User Agent Switcher, el cual permite al navegador Firefox simular el user-agent del dispositivo que se desee.

### **3.4.3 Análisis de la página de entrada al punto de acceso Wi-Fi**

Una vez definidas las categorías en las que clasificar a los móviles, el siguiente paso es rediseñar la página web original para cada una de las categorías, pero para ello es necesario analizar la página y sus elementos.

#### **3.4.3.1 Identificación de los elementos de la página**

Para poder determinar cómo debería verse una página en los distintos dispositivos móviles, se requiere conocer los bloques o elementos en los que puede dividirse la página original, para poder saber los que son indispensables, y por tanto deberán estar presentes en cualquier caso, y de los que se puede prescindir cuando se dispone de un espacio limitado, como es el caso de los móviles de peores características técnicas. Además, analizando la importancia de dichos elementos, se puede asignar a cada uno de ellos una determinada prioridad, la cual puede ser usada para especificar el orden en el que deberían ser mostrados los elementos para los distintos dispositivos, en los casos en que, debido a limitaciones de espacio o de cualquier tipo, se ha de modificar la disposición de los elementos de la página original.

Como resultado de dicho análisis, se obtiene que el elemento con mayor prioridad es la publicidad (tanto si es interna, en forma de cualquier imagen corporativa o logotipo, como externa, de cualquier socio de la compañía). Cabe reseñar que, a los ojos del usuario, la publicidad no es, ni mucho menos, el elemento más importante de una página web, pero desde el departamento de marketing, la publicidad sería vista como un elemento crucial, el cual debería ser lo primero que el usuario vea al acceder a la página web. No obstante, es de esperar que la publicidad o imagen corporativa no tenga una presencia demasiado significativa en términos de espacio, por lo que es admisible el disponer la publicidad en primer lugar, anteponiéndola a los elementos de la web que le dan a esta la verdadera funcionalidad.

El elemento siguiente, en grado de importancia, es el contenido en sí. Dado que el contenido es el elemento principal de la página, cabe hacer distinción entre las distintas partes



en las que puede subdividirse el contenido para cada página del hotspot. En el caso especial de la página de inicio del hotspot, la página del login, se pueden diferenciar dos bloques principales, como puede verse en la figura 14: un cuadro con información general acerca de las diferentes formas de convertirse en cliente del hotspot, así como de los distintos métodos de pago, y otro bloque en el que se puede escribir el nombre de usuario y la contraseña para iniciar sesión en el hotspot si ya se es cliente. En este caso concreto, el bloque para iniciar sesión tiene una prioridad más alta que el bloque con información, puesto que el objetivo de la página es que los clientes puedan iniciar sesión en el sistema, además de que desde un enlace del menú se puede acceder a una versión ampliada y más completa de la información para convertirse en cliente, lo que convierte a dicho elemento en información ciertamente redundante, y no estrictamente necesaria, para la página de inicio, en los casos en los que hay que ajustar el contenido, debido a limitaciones de espacio.

Después del contenido, el menú es el siguiente elemento a tener cuenta, el cual es posible analizar más profundamente, elemento a elemento, ya que permanece invariable para todas las páginas del hotspot.

En última instancia, el pie de página como elemento menos importante, el cual incluye información que puede ser obtenida a través de algún enlace del menú, como por ejemplo la información de contacto, y otras informaciones no muy relevantes, al menos en comparación con el resto de elementos de la página.

Como se ha advertido anteriormente, el menú puede ser analizado para conocer la importancia de sus elementos. Así, se tiene que los enlaces del menú pueden clasificarse en tres grupos por orden de mayor a menor importancia:

1. Los enlaces de mayor importancia son “Homepage – Login”, “Service plans” y “Help”, porque son los que llevan al usuario a la página para iniciar sesión, a la información tarifaria y a la ayuda.
2. Enlaces de menor importancia son “Hotspot Locations & Partners”, “Contact & feedback” and “Security”. Este último, debido a sus características, podría ser incluido dentro del enlace “Help”, ya que tanto el diseño como la semántica de ambos es similar.
3. Los enlaces menos trascendentales para los usuarios que visitarán la página de acceso al hotspot son “Downloads” y “My Hotspot”. La razón es que estos enlaces solo

resultan interesantes para los usuarios que son realmente clientes del hotspot, por lo que se podría posponer el mostrarlos hasta que el usuario haya accedido al sistema, tras introducir su nombre de usuario y contraseña, siempre que fuera necesario por motivos de ahorro de espacio.

Los elementos que se han ordenado por su grado de importancia durante el análisis de la página web pueden identificarse en la figura 14.

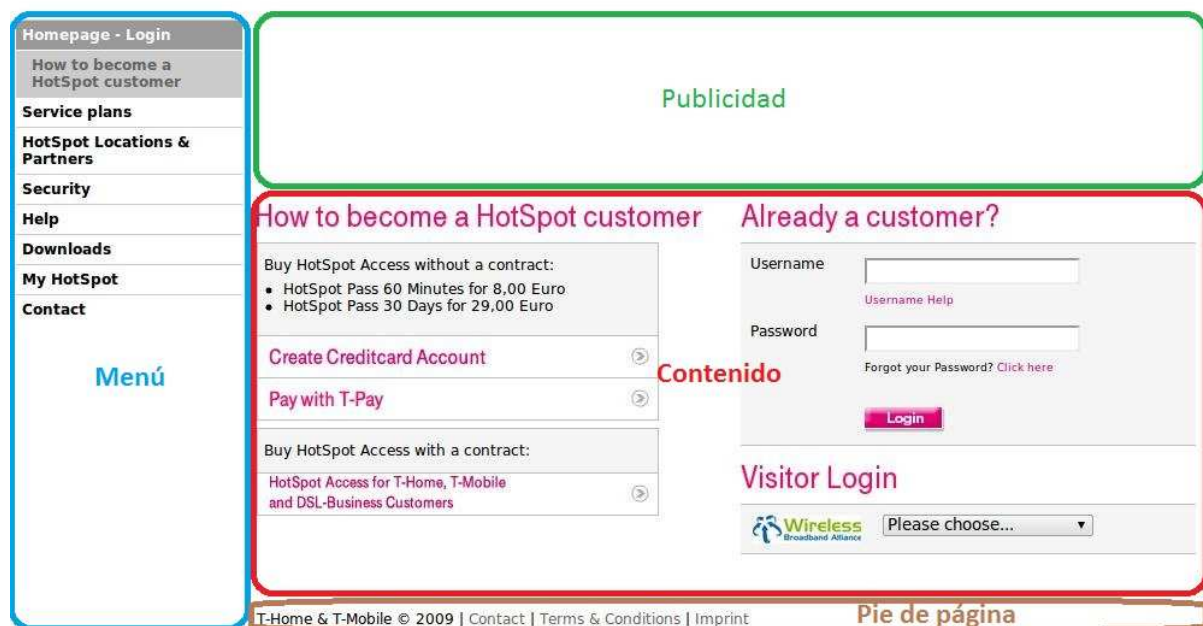


Figura 14: Elementos de la página original

### 3.4.3.2 Identificación de las tecnologías de las que hace uso la página

Durante el análisis de la página de acceso al hotspot se han analizado también las tecnologías de las que hace uso la página, para saber las posibles limitaciones de los dispositivos móviles a la hora de mostrar la web, ya que algunos, debido a sus reducidas características técnicas, en comparación con los ordenadores, no soportan determinadas tecnologías.

Así, se ha comprobado que la página del hotspot es bastante sencilla en cuanto a las tecnologías de las que hace uso. Fundamentalmente, la página está elaborada en HTML, con hojas de estilo CSS. En el código HTML aparecen algunos scripts simples en JavaScript, pero por otra parte, no se hace uso de Flash, ni de Java, lo que facilita la compatibilidad con cualquier móvil.

Es de destacar en este apartado que, la copia de la página utilizada para la realización de este proyecto ha sido la obtenida mediante la herramienta wget, la cual no es completamente compatible con JavaScript. Por esta razón, alguno de los scripts de la versión copiada de la página no funciona correctamente. No obstante, esto no afecta al objetivo del proyecto, el cual es adaptar la página original para su visualización en dispositivos móviles, aparatos de menores prestaciones y peores características técnicas que las de los ordenadores de sobremesa o portátiles.

Además, la página original del hotspot de la que se ha realizado la copia es la versión en inglés, la cual difiere de la versión en alemán, a simple vista, en un script, el cual no aparece en la versión en inglés. Dicho script es más complejo que el resto de scripts incluidos, por lo que podría dar más problemas a la hora de tener que adaptarlo para los móviles. Este script está contenido en un iframe, esto es, un elemento HTML que permite incrustar otro documento HTML dentro de la página HTML actual. En este caso, el elemento HTML consta de una cabecera fija y estática, y un contenido dinámico, que aparece en desplazamiento vertical continuo e infinito, en el cual se muestran varios titulares de noticias de índole económica.

#### **3.4.4 Rediseño de la página web para cada categoría**

En este punto, se tienen ordenados los elementos de la página de acuerdo a su importancia, por lo que se puede proceder a diseñar la página web adaptada para cada una de las categorías. Como una primera aproximación, y a modo de boceto, se hace un rediseño manual partiendo de la página original, modificándola cambiando el código HTML directamente, y también mediante el editor HTML KompoZer, el cual tiene la virtud de ser WYSIWYG (“What You See Is What You Get”, es decir “lo que ves es lo que obtienes”), con lo que se facilita la tarea de reubicar los elementos de la página, así como la de modificar sus características. De esta forma se tendrá una página HTML estática para cada categoría, que sirva de referencia para la creación de las páginas adaptadas y dinámicas más adelante, e igualmente se podrá implementar todo el sistema de detección del dispositivo y posterior redirección de éste a la página estática adaptada de su categoría, con el fin de comprobar la eficacia del mismo, mediante pruebas reales con móviles.

En primer lugar, cabe destacar que, para el diseño estático de las páginas adaptadas para las categorías, no se tuvo en cuenta el elemento publicidad. La razón es que, en la copia de la página original con la que se trabajaba dicho elemento aparecía en blanco, en lugar de aparecer alguna imagen corporativa. Esto era debido a algún problema de incompatibilidad de la herramienta utilizada para realizar la copia de la página, “wget”. Después de presentar el trabajo realizado hasta el momento (página estática adaptada) en una reunión con expertos, estos recomendaron la reserva de un espacio prioritario para publicidad, ya que sería un requerimiento por parte del departamento de marketing. De esta forma, tras la reunión, se incluyó manualmente una imagen publicitaria en la copia local de la página web, para que pudiera ser tratada durante el proceso de adaptación automática que obtiene la página dinámica final para cada categoría.

#### **3.4.4.1 Diseño estático para la categoría de paisaje**

Como se puede observar en la imagen de la figura 14, la página se ha modificado ligeramente para los dispositivos pertenecientes a la categoría de paisaje. La apariencia es bastante similar, simplemente se ha reducido su tamaño y realizado algunas modificaciones menores:

- Se comprueba que el ancho de la página original es de 980 píxeles, ya que está diseñada para una resolución de 1024 píxeles de anchura. En esta primera aproximación, en la que se realizará un rediseño con el que se obtendrá una página estática, se decide que la anchura de ésta página resultante para la categoría de paisaje sea de 570 píxeles. Por tanto, la reducción es de tres quintos, es decir, algo menos de la mitad, ya que  $570/980 \approx 0.6 = 3/5$ . Así, habrá que ajustar todos los elementos originales para su correcta visualización en el nuevo tamaño, modificándolos teniendo en cuenta la relación de tamaño de dos tercios.
- Se reduce ligeramente el tamaño de las fuentes. En este momento, se detecta una característica de la página original que genera un inconveniente a la hora del tratamiento del texto, más en concreto de los títulos. El problema está en los títulos de color rosa, y en los enlaces con texto rosa. Ambos son imágenes, con lo que si se quiere reducir su tamaño, de nada sirve el hacer más pequeña la fuente. En estos casos, habría que hacer más pequeñas las dimensiones de cada imagen, pero incluso con este sistema, los enlaces no quedarían bien dimensionados, puesto que el espacio que separa el texto del enlace de la flecha de la derecha podría ser demasiado grande, pero al pertenecer todo a la misma imagen no se puede modificar. Esto será una limitación

para la adaptación automática futura. En esta etapa del proyecto se ha solucionado esto modificando las imágenes manualmente usando un programa de edición de imágenes, GIMP, para eliminar parte del espacio de separación, como puede verse en las imágenes de la figura 15.



Figura 15: Imagen original y modificada manualmente con GIMP

Como se puede advertir en la figura 16, la página web original aparece dividida verticalmente en dos partes, ocupando el menú la parte izquierda, y el contenido en sí la parte derecha. Ambas partes han visto reducido su tamaño, sobre todo la de la derecha, que ha pasado de tener el contenido organizado en dos columnas a tenerlo estructurado en una sola de las columnas. La parte del menú también ha visto reducido su espacio para ajustarse al nuevo tamaño de la web.

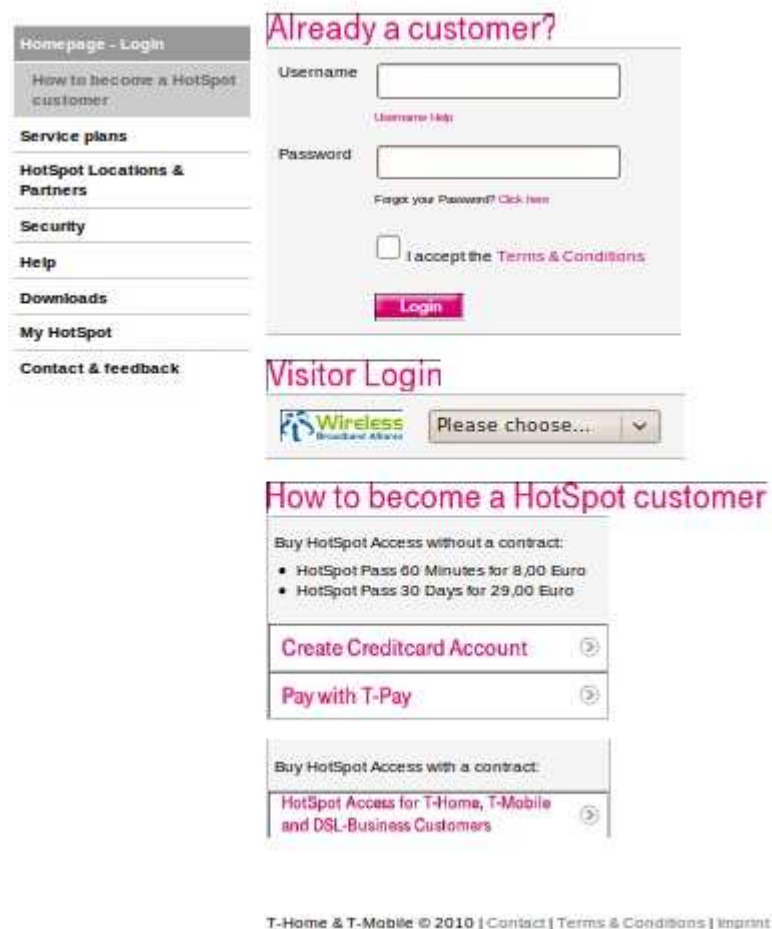


Figura 16: Diseño estático para la categoría de paisaje

Una parte vital es la referente a la nueva estructura del contenido. De las dos columnas de las que constaba el contenido, la de la derecha, titulada “Already a customer?” que sirve para que los usuarios puedan escribir su nombre de usuario y contraseña, ha pasado a ser la primera, mientras que la de la izquierda se muestra a continuación debido a su menor importancia, discutida anteriormente en el apartado de las prioridades de los elementos de la página web original. Por otra parte, ambos elementos del contenido han visto reducir sus dimensiones, ajustando los bordes al texto que contienen.

Debido a que los dispositivos de la categoría de paisaje tienen altas características técnicas, no se ha creído conveniente eliminar ninguno de los elementos de la página web original, para que los usuarios tengan una experiencia de usuario más cercana a la conseguida con la página original.

#### **3.4.4.2 Diseño estático para la categoría de retrato**

Se puede ver en la imagen de la figura 17 la apariencia de la página web estática adaptada para los dispositivos de la categoría de retrato. En este caso, la apariencia del sitio web adaptado difiere bastante de la de la página original, aunque los colores corporativos y la estructura de los elementos se mantiene. Los elementos de la página han sido reubicados, y la estructura es completamente distinta, buscando el ajuste perfecto a los dispositivos pertenecientes a esta categoría. Estas son las modificaciones realizadas a la web original:

- Para la categoría de retrato se opta por rediseñar la página original a una anchura de 240 píxeles para que ajuste bien a la mayoría de dispositivos de esta categoría, lo que implica reducir considerablemente el tamaño original de 980 píxeles. Así pues, la página adaptada para esta categoría será cuatro veces más pequeña que la original, lo que da una idea de la magnitud de las modificaciones que se han realizado.
- Al igual que con la categoría de paisaje, el tamaño de las fuentes se ha reducido ligeramente, sin hacerlas excesivamente pequeñas, para que fuera todavía posible leer los textos. El problema de los títulos que son imágenes también se ha tenido a la hora de adaptar manualmente la página para esta categoría.

Como se puede comprobar en la imagen de la página adaptada de la figura 17, en esta categoría ha habido que cambiar completamente la estructura de la página debido a las limitaciones de tamaño impuestas por los móviles pertenecientes a la categoría. Así, la

estructura del sitio web original, y de la página adaptada para la categoría de paisaje, en la que se podían apreciar dos zonas que dividían la página verticalmente, para incluir el menú en la parte de la izquierda y el contenido en la parte de la derecha, da paso a una sola columna en la que aparecerán todos los elementos de la web, unos a continuación de otros.

The image displays a static design for a website, organized into a single column. At the top, there is a section titled "Already a customer?" in pink. Below this title is a login form with two input fields: "Username" and "Password". The "Username" field has a small "Username Help" link below it. The "Password" field has a "Forgot your Password?" link below it. Below the password field is a checkbox labeled "I accept the Terms & Conditions" and a pink "Login" button. Below the login form is a section titled "Visitor Login" in pink. This section features the "Wireless Broadband Alliance" logo on the left and a "Choose..." dropdown menu on the right. Below these sections is a vertical menu with the following items: "Homepage - Login", "How to become a HotSpot customer", "Service plans", "Help", "Security", "HotSpot Locations & Partners", and "Contact & feedback". At the bottom of the page, there is a copyright notice: "T-Home & T-Mobile © 2010".

**Figura 17: Diseño estático para la categoría de retrato**

Como consecuencia de la nueva estructura de la página, en la que todos los elementos aparecen en la misma columna, si se mostraran todos los elementos, la longitud de la página podría considerarse excesiva para los terminales en los que será visualizada. Por ello hay que decidir de qué elementos se puede prescindir, de acuerdo al análisis de los elementos de la página y su importancia, que ha sido explicado en su apartado correspondiente. De esta forma, se opta por suprimir la parte izquierda del contenido, la titulada "How to become a HotSpot customer", debido a que la información que detalla puede ser obtenida a través de un enlace del menú, y no es de tanta importancia como el resto del contenido. Por otra parte, ciertos enlaces del menú han sido igualmente eliminados para reducir la longitud de la página

adaptada, como son “Downloads” y “My HotSpot”. La información obtenida a través de estos enlaces no es de interés para un usuario de la página que no se ha conectado al hotspot, solo tienen sentido para usuarios reales del hotspot, por lo que podrían aparecer en el menú cuando el usuario introduzca su nombre de usuario y contraseña. Además, también se han eliminado los enlaces del pie de página, debido a que son redundantes, porque se puede acceder a los mismos a través de enlaces del menú, dejando en el pie de página solo la información referente al copyright de la misma.

Los elementos del menú han sido reorganizados siguiendo las indicaciones del apartado de la importancia de los elementos de la página. Así el elemento “Hotspot Locations & Partners” aparece retrasado, y se ha adelantado la posición del elemento “Help” con respecto al elemento “Security”.

El nuevo diseño de la página adaptada para la categoría de retrato consta de una sola columna, como se ha especificado anteriormente. Todos los elementos se han ajustando a la anchura anteriormente especificada de 240 píxeles, por lo que han visto reducir su tamaño, y el de los sub-elementos que los componen. El primer elemento en aparecer es el contenido, de importancia máxima, ya que es el que permite al usuario conectarse al punto de acceso. Seguidamente aparece el menú, siendo el pie de página el último elemento, como es normal. Esta nueva distribución es bastante ajustada, considerando que las características técnicas de los dispositivos de esta categoría no son generalmente tan avanzadas como las de los dispositivos de la categoría de paisaje. No obstante, esta reubicación de los elementos es una solución elegante que hará sentirse cómodos y satisfechos a los usuarios que usen el sitio web adaptado, debido a la conservación del aspecto de la página original, en cuanto a colores corporativos, diseño del contenido y del menú, etc. Asimismo, la longitud de la página no es excesiva, debido a la ocultación de los elementos no esenciales, evitando así al usuario el tener que realizar un desplazamiento vertical excesivo para buscar la información que desea, puesto que ésta está más contenida.

### **3.4.5 Adaptación automática**

En este punto, se tienen dos páginas estáticas adaptadas, una para cada categoría, y se ha comprobado la eficacia del sistema mediante pruebas en laboratorio con dispositivos reales. Es por ello que, para continuar con el proyecto, se debe tomar como ejemplo las páginas estáticas adaptadas y desarrollar un sistema que, tomando la página original, obtenga



dichas páginas adaptadas automáticamente, para después dotar a dichas páginas de un comportamiento dinámico. Por lo tanto, el siguiente paso es realizar la adaptación de la web propiamente dicha. Esta adaptación pretende ser realizada al vuelo automáticamente, es decir, cuando el usuario móvil hace la petición de la página web, en lugar de ofrecerle la página directamente, se detecta que es un dispositivo móvil y se realiza la operación de adaptar la página original para una visualización óptima en el dispositivo, de acuerdo a las características del mismo obtenidas en el momento de la detección. Así, como respuesta a la petición original, se sirve el producto de la adaptación al usuario, el cual no aprecia que ninguna operación de ajuste esté llevándose a cabo.

La forma de adaptar la página web es, a través de un script PHP, ir ajustando todos los elementos (menú, contenido...), uno a uno, en el propio código HTML. No se pueden hacer transformaciones directas de los elementos, porque los elementos lógicos están delimitados por elementos HTML del tipo contenedor (div), y no se pueden transformar estos divs directamente por la pérdida de semántica de la página que implicaría, ya que en muchos casos hay divs anidados unos dentro de otros, y se hace necesario seguir la estructura de estos. Por esta razón se ha de utilizar un analizador sintáctico DOM (Document Object Model).

En vez de utilizar la extensión DOM de PHP directamente, se ha decidido utilizar el analizador sintáctico “Simple HTML DOM” debido a su facilidad de uso. Aunque no tiene el potencial de la extensión DOM de PHP, ya que no incorpora toda su funcionalidad, es suficiente para lo que se necesita en el proyecto, de hecho, tiene funciones más complejas de las que van a ser utilizadas.

Para los objetivos de este proyecto, el analizador sintáctico DOM lo que hace es leer la página HTML original, y crear en memoria todos los elementos de los que consta la página original, en forma de objetos, con estructura de árbol, para conocer la relación de parentesco que tienen todos los elementos HTML entre ellos, o dicho de otro modo, qué elementos están dentro de qué elementos. De esta forma, se puede tratar directamente cualquier elemento de la página HTML original para modificarlo, borrarlo, o crear nuevos elementos, todo ello muy fácilmente, ya que el analizador DOM ofrece una sintaxis sencilla para realizar todo tipo de acciones. Para apreciar las ventajas de utilizar DOM, se puede observar que si no se utilizara, y se quisiera modificar algún elemento, habría que buscar secuencialmente el elemento HTML en la página realizando comparaciones de texto, ya que la página HTML es solamente

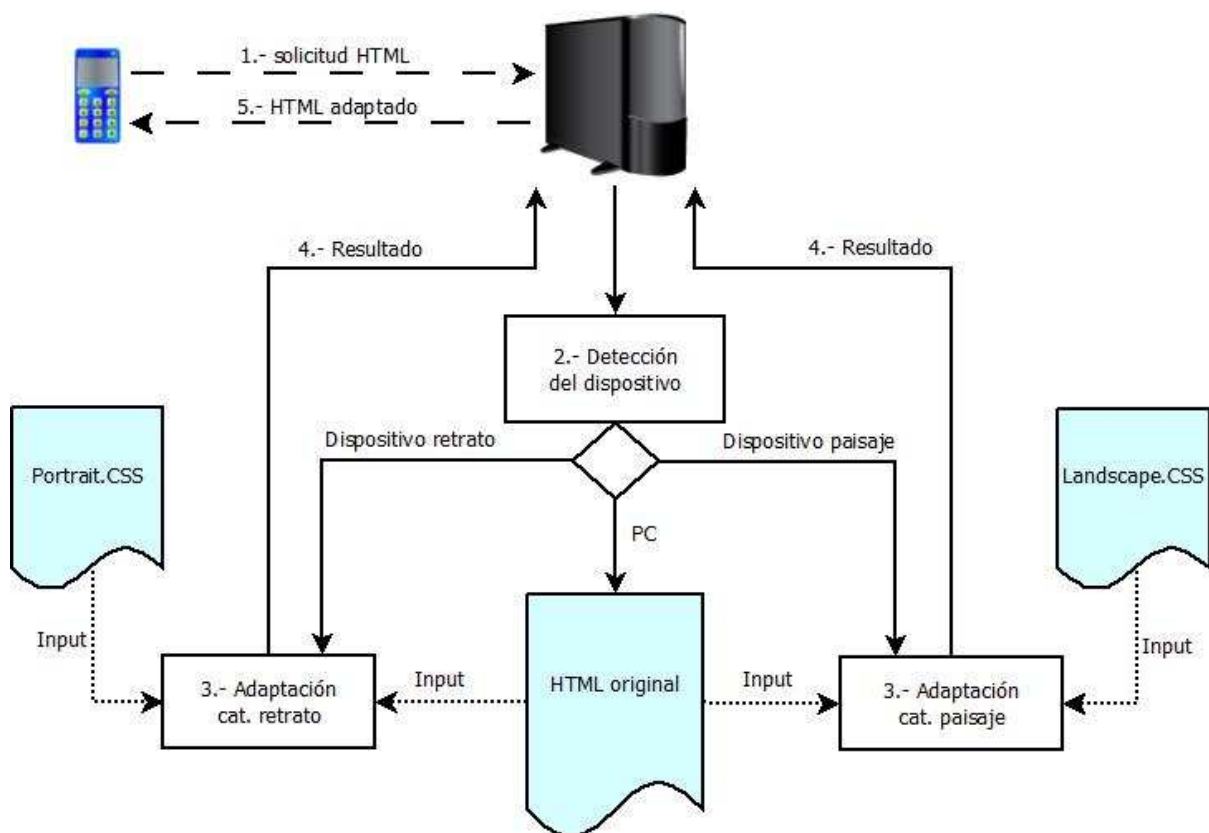
un fichero de texto plano, lo que implicaría un código de un nivel extremadamente bajo, y con poca lógica, lo que haría muy complicado su entendimiento.

De esta forma, el analizador DOM se utiliza en primer lugar para leer la página original, y recrearla en memoria. A continuación, se realizan todas las modificaciones que se crean oportunas en la estructura de la página web que el analizador DOM tiene en memoria, para, finalmente, volcar el contenido de la estructura DOM en memoria, con lo que se visualizará la página ya adaptada en el dispositivo correspondiente. Otra de las ventajas es que, con este volcado, se evita tener que hacer redirecciones, lo que implicaría un aumento de tiempo de espera para el usuario. Por otra parte, si en cualquier momento se decide tener el resultado de la adaptación como una página guardada en memoria para ahorrar tiempo, y no tener que realizar la misma adaptación a un dispositivo, una y otra vez, el analizador DOM permite salvar la estructura que tiene en memoria a un fichero HTML, en lugar de volcarlo directamente como se hace en el proyecto.

Durante el uso de la estructura de la página en DOM se ha apreciado algo que es necesario tener en cuenta para futuras modificaciones. El caso problemático es cuando se desea modificar un elemento que contiene otro elemento anidado que también se quiere modificar. En esos casos, hay que modificar en primer lugar el elemento hijo. Si se modifica, en primera instancia, el elemento padre, no sería posible modificar el hijo, ya que el propio árbol DOM no sería capaz de encontrar el elemento hijo, a no ser que antes se volcara el árbol DOM en un fichero, y se volviera a cargar. La razón es sencilla: en el momento en que se modifica el elemento padre, el árbol DOM desconoce si lo que hay contenido en el elemento padre es otro elemento o no, ya que ha habido una modificación, y el árbol DOM solo puede conocer la semántica de los elementos, y sus relaciones, en el momento en que se lee la página web y se crea el árbol.

En el momento en que el dispositivo móvil recibe el volcado del árbol DOM, con el resultado de la página adaptada, como ya se explicará en los siguientes apartados, ésta necesitará una nueva hoja de estilos CSS. Esta hoja de estilos CSS, una para cada categoría, no forma parte del proceso automático, puesto que se han modificado manualmente, partiendo de la hoja de estilos original, y están almacenadas junto a ella. Debido al objetivo de automatizar todo el proceso de adaptación, se intentó originalmente adaptar automáticamente la propia hoja de estilos CSS, al igual que se hace con la página HTML, para así obtener las

hojas de estilos adaptadas desde la hoja de estilos original. No obstante, también sería necesario utilizar un analizador sintáctico, como se hace en el caso de la página HTML con el DOM, pero no se encontró ninguna herramienta que permitiera realizar lo que se requería. Entonces, se planteó la posibilidad de desarrollar una herramienta que resolviera el problema, pero el tratamiento de toda la hoja de estilos no es un asunto trivial, ya que dicha herramienta debería ser capaz de reconocer todos los posibles elementos de una hoja de estilos, es decir, habría que definir toda la gramática. En fin, habría que desarrollar un analizador sintáctico, y esto requiere demasiado esfuerzo, teniendo en cuenta el tiempo fijo de desarrollo del proyecto.



**Figura 18: Diagrama del proceso completo de detección del dispositivo y adaptación de la página**

Por estas razones, se decidió modificar manualmente la hoja de estilos CSS original, para obtener dos versiones distintas, una para cada categoría, como se puede apreciar en el diagrama de la figura 18. Esta aproximación tiene un punto débil, y es que, si desde la compañía decidieran modificar la hoja de estilos, habría que modificar las otras hojas de estilos también, lo cual es redundante. Sin embargo, en el caso de que se desarrollara un analizador para usarlo en la adaptación automática de la hoja de estilos, y por parte de la compañía, se realizara algún cambio en la hoja de estilo, sería necesario tener estos cambios

en cuenta y podría tenerse que modificar también el script de adaptación, ya que este está preparado para la hoja de estilos actual. Por eso, en caso de modificación, habrá que cambiar algo, independientemente del sistema finalmente empleado. Por otra parte, cabe destacar que los cambios en la hoja de estilos son muy infrecuentes, hecho advertido por los propios responsables de la página del hotspot, por lo que no es necesario tomar decisiones basándose únicamente en la posibilidad de que ocurra alguna modificación, si bien es una buena práctica el tener en cuenta dicha posibilidad.

#### **3.4.5.1 Requisitos de la página web original**

En primer lugar, hay que recordar que la copia de la página original con la que se trabaja fue obtenida con la herramienta wget, la cual no es completamente compatible con JavaScript, y posiblemente por esto, la zona de la página donde aparece el logotipo de la compañía, y alguna otra imagen, aparecen completamente en blanco en la copia. Para poder trabajar con una página más ajustada a la realidad, se ha añadido una imagen en la cabecera de la página, a modo de publicidad, para poder tratar este elemento como tal. En concreto, el elemento HTML contenedor en el que se ha puesto la imagen, ha sido en el div “header” directamente, elemento que estaba vacío en la copia local de la página, debido a los problemas anteriormente expuestos.

Para realizar la adaptación se ha buscado ser mínimamente intrusivo, de forma que, idealmente, no hubiera que hacer ningún cambio en la estructura de la página HTML original. No obstante, ha sido necesario realizar algún cambio para un correcto, y lógico, funcionamiento de los scripts que realizan la adaptación de la página original. A continuación se listan los requisitos que debe satisfacer la página original, así como las razones por las que se han tomado estas decisiones:

- Todos los elementos HTML contenedores (div) han de tener un atributo id que les identifique, o al menos todos los que vayan a ser modificados. Este hecho, que podría resultar obvio, no lo es tanto, ya que en la página HTML original el pie de página aparece representado como un elemento div que pertenece a la clase “footer”, pero no tiene id, por lo que este es uno de los cambios que se han realizado en la página HTML.
- La estructura original de la página es demasiado general. Los elementos de los que consta son: publicidad, menú, contenido y pie de página. Si se quiere realizar una buena adaptación, se requiere que en el elemento “content” se reconozcan distintos

sub-elementos. Por esta razón, para la página principal del hotspot se han añadido los elementos “login”, que representa al bloque para introducir el nombre de usuario y contraseña, y “become”, que identifica al bloque en el que aparece información general acerca de cómo convertirse en cliente del hotspot, y de las tarifas disponibles. Disponiendo de estos divs adicionales, se podrá cambiar la posición de los elementos, e incluso ocultar alguno de ellos, dependiendo de las características del dispositivo que haya hecho la petición de la página. Así, para el resto de páginas del hotspot se requeriría que se estructurara internamente el elemento “content”, clasificando el contenido en nuevos divs.

- En la hoja de estilos CSS original, los tamaños de los elementos están definidos de forma absoluta, es decir, en píxeles. Sin embargo, si se quiere que la adaptación sea dinámica, esto es, que se ajuste al dispositivo móvil sea cual sea, teniendo en cuenta sus características de visualización, se debe definir la anchura de todos los elementos en el fichero CSS de forma relativa, es decir, en porcentajes. Así pues, cada elemento ocupará un porcentaje de la anchura del elemento en el que está contenido. Solo habrá una excepción, un elemento que englobe los demás elementos que muestren algo por pantalla. Este deberá ser definido de forma absoluta, especificando los píxeles que deba ocupar. A partir de ese tamaño fijo, sus descendientes se ajustarán de acuerdo a los distintos porcentajes que tengan definidos en el atributo anchura. El elemento base por el que se ha optado es “layer\_middle”. Podría parecer que el hecho de especificar una anchura fija por defecto para el elemento raíz, implica que la anchura de los elementos mostrados va a ser la misma para cualquier dispositivo que acceda a la página, pero no es así ya que, una vez detectado el dispositivo, se obtendrá su resolución nativa de pantalla, y se usará este valor para sobrescribir, al vuelo, el valor estático del elemento “layer\_middle” mediante un estilo inline. De esta forma, se obtiene un comportamiento dinámico, en el que la página va a ajustarse completamente a la pantalla del dispositivo. En cualquier caso, se ha tenido que decidir una anchura mínima para la versión adaptada de la menor categoría (“portrait”) ya que la visualización de una página demasiado estrecha no es cómoda para el usuario, porque, por ejemplo, puede llegar a mostrar un simple párrafo a razón de 1 ó 2 palabras por línea, haciendo las páginas excesivamente alargadas verticalmente. Así, para los móviles con una anchura menor que 200 píxeles, la página adaptada resultante tendrá en todos los casos un ancho de 200 píxeles.

- Todos los elementos de la página HTML deben obtener sus formatos de la hoja de estilos CSS correspondiente, y nunca sobrescribir los valores de estos mediante definiciones de estilos inline. Se ha comprobado que en la página original existen elementos con definiciones de estilos inline, como imágenes y tablas. Esto hace que dichos elementos ignoren por completo el tamaño que deberían tener de acuerdo al fichero CSS.
- Se debe tener una hoja de estilos CSS diferente por cada diseño distinto de la página adaptada que se quiera tener. En este caso se necesitará un fichero CSS por categoría, ya que por ejemplo, para la categoría “portrait”, todos los elementos deberán aparecer ocupando todo el ancho de la pantalla, es decir, tendrán asignado en el atributo “width” un valor del 100%, mientras que, para la categoría “landscape”, el menú aparecerá a la misma altura que el contenido, de forma que entre los dos elementos sumarán 100%, ocupando concretamente el menú un 33% del espacio, y el contenido el restante 67%.

#### **3.4.5.2 Atributos para controlar la futura adaptación**

Aparte de estos requisitos, se han tomado una serie de decisiones, en cuanto a la inclusión de nuevos atributos en la página HTML original, para que el editor de esta pueda controlar cómo serán las versiones adaptadas para los móviles, modificando los valores de nuevos atributos, en el momento de creación del HTML para ordenadores de sobremesa:

- Los elementos div de la página, así como los elementos de listas como por ejemplo los enlaces del menú principal, pueden tener un nuevo atributo, llamado “*categoría\_hide*”, donde *categoría* puede ser “portrait” o “landscape”. Los valores que puede tomar este atributo son “yes” y “no”. Así, cuando este atributo aparezca con el valor “yes” en un elemento, el script de adaptación ocultará dicho elemento. En el resto de los casos, cuando aparezca con el valor “no” o no aparezca, el elemento será mostrado normalmente. Esta decisión ha sido tomada después de observar que la información mostrada en los ordenadores de sobremesa es, en ocasiones, excesiva para un dispositivo móvil, en el que se debería ahorrar espacio, ocultando información que no sea imprescindible para el usuario móvil.
- No todos los elementos de la página tienen la misma importancia. Y no todos los elementos tienen por qué aparecer en el mismo orden en los ordenadores de sobremesa que en los móviles. Por esta razón, se ha tomado la decisión de incluir la funcionalidad

de que los elementos puedan ser reorganizados, de forma que, para las versiones adaptadas para los móviles, los elementos puedan aparecer en distintas posiciones de acuerdo a la importancia de los mismos. Para poder reordenar los elementos, estos han de incluir un atributo llamado “priority”, el cual representará la prioridad del elemento con respecto a los demás elementos mediante un número, siendo el “1” el elemento con mayor prioridad.

- En relación con el apartado anterior, y teniendo en cuenta lo que se mencionó anteriormente en los requisitos respecto a que el elemento “content” debería tener sus sub-elementos identificados y bien definidos, se ha decidido añadir el atributo “content\_priority” a los bloques en los que se subdivide “content”. Dicha prioridad se circunscribe al contexto de los sub-elementos del elemento “content”. De esta manera se pueden reorganizar dichos sub-elementos de forma independiente a la reordenación explicada en el punto anterior, pero con un funcionamiento idéntico, lo que implica que, paralelamente, el elemento perteneciente al contenido que tenga una mayor prioridad será el que aparezca en primer lugar dentro del ámbito del contenido.

En cuanto a estos nuevos atributos de los elementos que la página original debe incluir, si se pretende que la adaptación funcione correctamente, durante la realización del proyecto se observó que, dado que estos nuevos atributos eran añadidos manualmente a la copia de la página original con la que se trabaja, resultaba una ardua tarea el hacerlo, puesto que se debía añadir dichos atributos en todas las páginas HTML, y no solo en la de inicio, si se quieren adaptar todas, como se explicará en un apartado posterior. No obstante, hay que pensar que el editor de la página original, cuando la diseñe, utilizará algún programa para ello, por lo que podrá tener automatizada dicha tarea, para que cuando añada un elemento se le solicite que defina su prioridad, y si va a ser escondido para alguna categoría. De esta forma el programa creará dicho elemento con sus atributos no solo para la página de inicio, como se ha ido haciendo originalmente en este proyecto, sino para todas las páginas en las que aparezca el elemento, por lo que no implica un gran esfuerzo por parte del editor, solamente implica tener en mente que la página que se está diseñando para los ordenadores, va a ser adaptada automáticamente en el futuro para su óptima visualización en dispositivos móviles, y el proceso de adaptación automática necesita ciertos atributos para saber los cambios que tendrá que realizar a la página original.

En versiones anteriores del script de adaptación, la página web era reorganizada de forma distinta. En lugar de basar la reordenación de los elementos en prioridades definidas como atributos en la página HTML original, se utilizaba el identificador de los elementos div. Esta forma de reorganizar los elementos de la página web, a pesar de ser más clara, puesto que se indica qué elemento va a continuación de qué elemento, es menos general, ya que hace depender a la página original de los actuales nombres de elementos. Con la nueva forma basada en asignar una prioridad a cada elemento, el script de adaptación es menos visual, pues no se aprecia a simple vista la nueva organización de los elementos, lo único que se podría advertir es, por ejemplo, que el elemento de prioridad 2 va a ir a continuación del elemento de prioridad 1.

Como mejora del sistema actual, se podría automatizar el proceso de reorganización, sin que hubiera necesidad de incluir sentencias en el código del script que permitan organizar los elementos uno a uno, sino una sola función que, sin saber de antemano el número de elementos a reorganizar, los reordene mediante un algoritmo de ordenación. Esta mejora se expondrá en el capítulo de Conclusiones y futuros trabajos de la presente memoria.

Otra funcionalidad que fue implementada para un paso intermedio en la elaboración del script final es la posibilidad de modificar el estilo de un elemento en tiempo de ejecución, esto es, añadiendo un estilo inline. De esta forma se puede cambiar, para cualquier elemento, cualquier atributo definido por defecto en la hoja de estilos CSS. De hecho en la versión final del script se utiliza una vez esta función, para sobrescribir el valor por defecto de la anchura del contenedor global, esto es, inyectando dinámicamente la anchura del dispositivo que está haciendo la petición, para que todos los elementos de la página web se ajusten al dato correcto. La anchura del dispositivo es obtenida por el módulo de detección del dispositivo de la base de datos WURFL, y es pasada, por URL, al módulo de adaptación de la página.

Para una mejor visualización de la página adaptada en los iPhone, dado que estos representan un alto porcentaje de usuarios móviles del hotspot de Deutsche Telekom, el script de adaptación añade, al resultado final de la web adaptada, una etiqueta meta que el resto de dispositivos móviles ignora por completo. El nombre de esta etiqueta es “viewport”, y con ella se puede hacer que el navegador del iPhone, Safari, muestre la página totalmente ajustada a la pantalla del dispositivo. De otra forma, Safari daría por hecho que la pagina web adaptada tiene un tamaño de página web estándar de 960 píxeles de ancho, pero el ancho real de la



página sería de 320 píxeles, por lo que la página aparecería en pequeño, como alejada, con lo que se obligaría al usuario a tener que hacer zoom para poder visualizar la página correctamente y esto no es óptimo ya que, al obtenerse la página tras un proceso de adaptación, se entiende que esta página debe poder ser visualizada sin tener que aumentar o alejar el zoom. Además esta etiqueta permite ajustar la posibilidad de que el usuario haga zoom en la página web. En este caso se ha deshabilitado dicha función, ya que realmente no es necesario realizar zoom a la página resultante, ya que ha sido adaptada específicamente para ese dispositivo. De esta forma, se tiene un mayor control sobre lo que el usuario está visualizando en todo momento. A continuación se muestra la sentencia añadida, en la que device-width es una variable con el valor de la anchura del dispositivo:

```
<meta name = "viewport" content = "width=device-width; initial-scale=1.02; maximum-scale=1.02; user-scalable=0;" />
```

### 3.4.5.3 Resultado de la adaptación automática dinámica para la categoría de paisaje

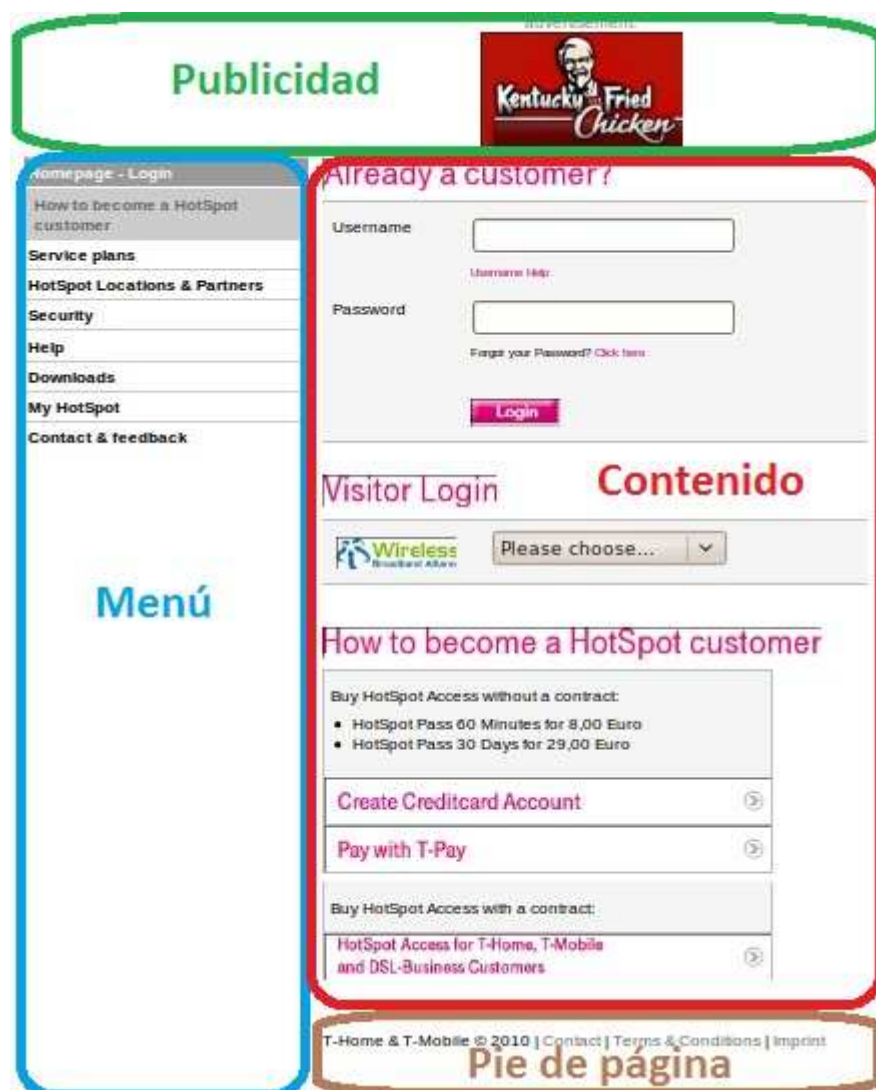
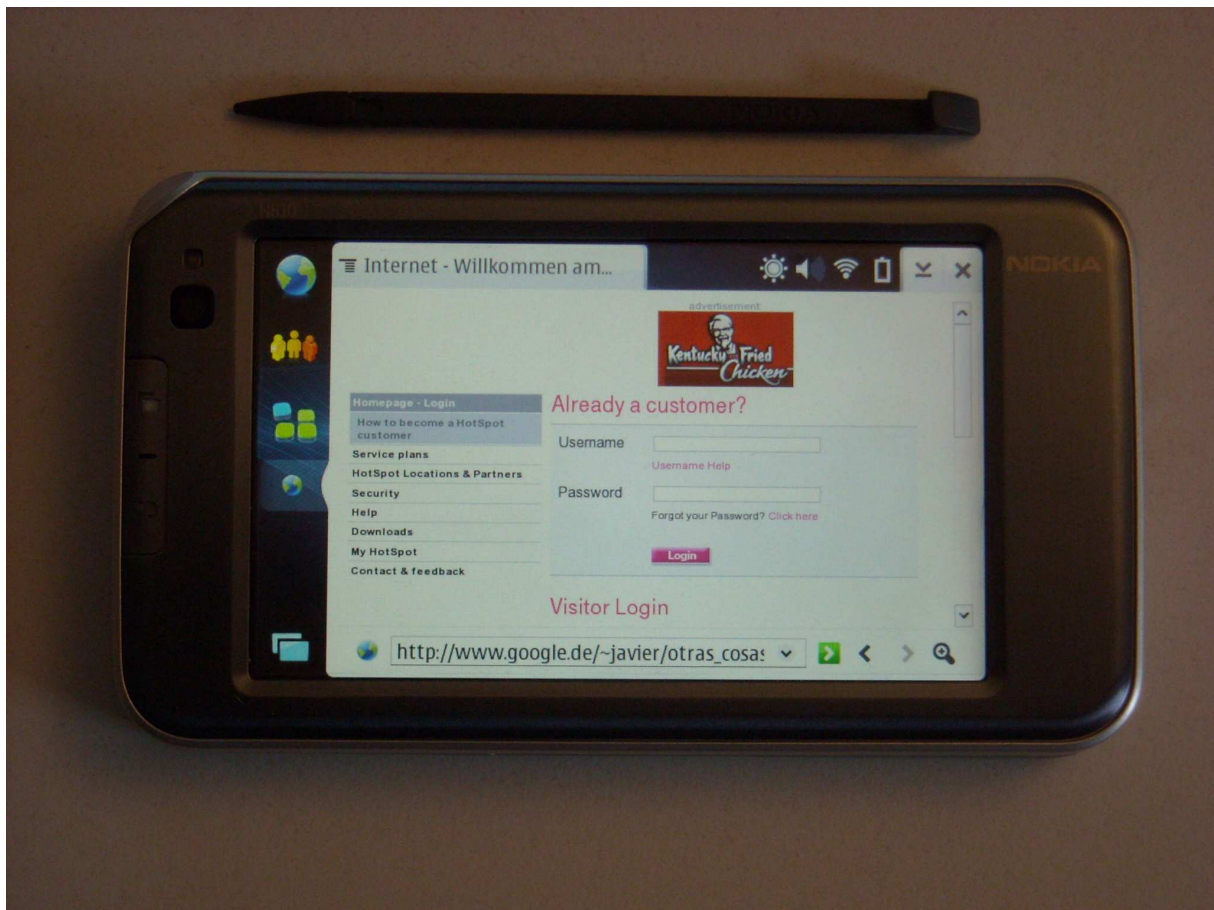


Figura 19: Disposición de los elementos tras la adaptación automática para la categoría de paisaje

De acuerdo al diseño estático para esta categoría explicado en el apartado correspondiente de esta memoria, y tras implementar todas las modificaciones explicadas anteriormente para proceder a adaptar automáticamente la página web, en la figura 19 se puede ver el aspecto que tiene la página adaptada para la categoría de paisaje.

Se puede observar que la publicidad tiene prioridad máxima, apareciendo al principio de la página. A continuación, el menú ocupa la parte de la izquierda de la página, mientras el contenido aparece a la derecha. Es de destacar que el contenido de la página inicial ha sido dividido, y se ha asignado una prioridad distinta a cada una de las partes, para así mostrar en primer lugar las cajas de texto donde deben escribir los usuarios su nombre de usuario y contraseña para conectarse al punto de acceso Wi-Fi. En último lugar, aparece el elemento de menor prioridad, el pie de página.



**Figura 20: Resultado de la adaptación automática en un Nokia N810 (categoría de paisaje)**

Se puede apreciar el resultado en un entorno real, es decir, utilizando un dispositivo que pertenezca a esta categoría, en la imagen de la figura 20. En concreto se ha utilizando un

Nokia N810, dato importante, ya que la página resultante es dinámica, es decir, se adapta completamente al dispositivo que la ha solicitado.

#### 3.4.5.4 Resultado de la adaptación automática dinámica para la categoría de retrato

Teniendo en cuenta el rediseño estático al que fue sometida la página para la categoría de retrato, que ya ha sido explicado, y siguiendo las modificaciones anteriormente expuestas para adaptar la página original, el aspecto que tiene la página adaptada para la categoría de retrato es el que se puede ver en la figura 21.



Figura 21: Disposición de los elementos tras la adaptación automática para la categoría de retrato

Todos los elementos han sido organizados en una sola columna, por orden de prioridad. En primer lugar, con prioridad máxima, aparece la publicidad, seguida del contenido, en el cual se pueden apreciar los efectos de las modificaciones automáticas, ya que

ha desaparecido una de las dos partes del contenido de la página inicial original, para dejar solo la parte en la que los usuarios pueden introducir su nombre de usuario y contraseña para conectarse al hotspot. Después aparece el elemento menú, en el cual se han eliminado automáticamente alguno de sus elementos, de acuerdo a los nuevos atributos definidos en la página inicial. Y finalmente, en último lugar, se encuentra el pie de página, con menor prioridad que los demás.



**Figura 22: Resultado de la adaptación automática en un Nokia E60 (categoría de retrato)**

Para observar el resultado más ajustado a la realidad, y apreciar su eficiencia, se añade a continuación, en la figura 22, una imagen real de un dispositivo que pertenece a la categoría

de retrato mostrando dicha página adaptada. En este caso en concreto se ha utilizado un Nokia E60.

En el apéndice B de la memoria se pueden apreciar los resultados para más dispositivos móviles, así como la visualización de la página en dichos móviles antes de realizar la adaptación, para poder apreciar las modificaciones realizadas.

### **3.4.5.5 Adaptación automática de todas las páginas de la web**

En principio, se ha trabajado a fondo en la adaptación de la página de entrada al punto de acceso Wi-Fi, en la que aparecen las cajas de texto en las que escribir el nombre de usuario y la contraseña para conectarse, pero desde ésta, los usuarios, pueden acceder a otras páginas a través del menú o de los enlaces, por ejemplo la página de ayuda, la de contacto, etc. De esta forma, lo ideal es que la adaptación sea realizada a todas las páginas. El problema radica en las limitaciones que se han ido exponiendo en la memoria, por ejemplo, con las imágenes, con las que habría que cuidar mucho la forma de tratarlas, para evitar que los resultados de la adaptación se vean limitados por ellas. Es por eso que las páginas más simples, como son la de ayuda o la de seguridad, en las que solo hay texto, la adaptación a los dispositivos es perfecta, y parece que ha sido expresamente diseñada para ellas, lo que demuestra la eficiencia de la solución propuesta para este proyecto.

Como se ha dicho, a pesar de haberse centrado en la página inicial del hotspot, al final del proyecto se trabajó en conseguir alguna forma de adaptar todas las páginas. Esto se ha conseguido utilizando un fichero .htaccess, que es un fichero de configuración de Apache que permite definir ciertas directivas, en este caso, reescribir la URL. De esta forma, el dispositivo realiza una petición de una página web HTML al servidor, pero éste, actuando de acuerdo al fichero .htaccess, sobrescribe dicha URL por la dirección del script de adaptación, pasándole por URL como una variable la página que el dispositivo ha solicitado, para que así el script de adaptación pueda adaptarla. El aspecto del fichero .htaccess que se ha utilizado es el siguiente:

```
RewriteEngine On
RewriteRule ^(.+)\.html$ /~javier/wurfl-php-1.r1/examples/demo2/index.php?file=%{REQUEST_URI}
[L,NC,QSA]
```



A continuación, en la figura 23, se incluye un esquema del proceso que se sigue con el fichero .htaccess.

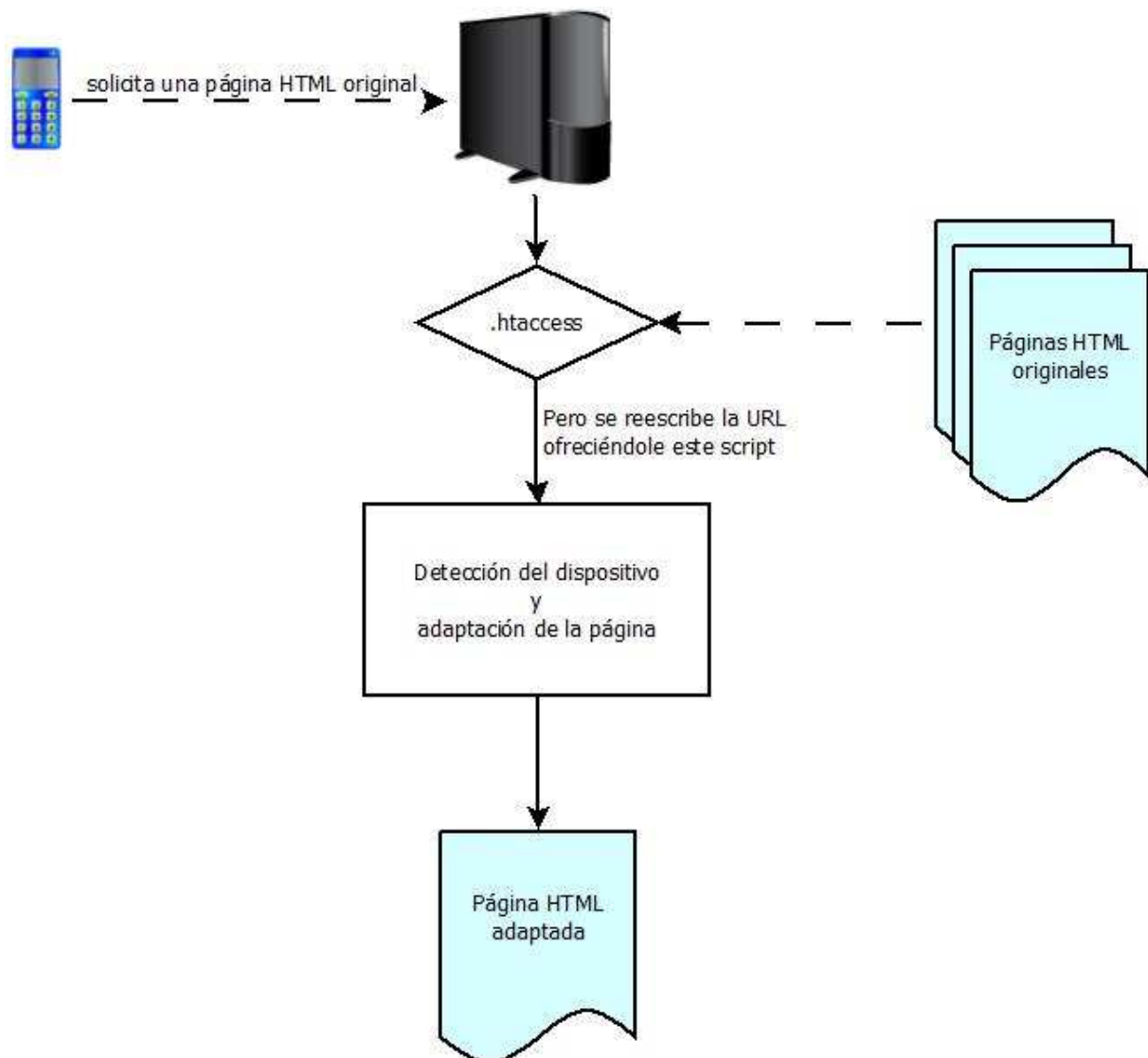


Figura 23: Diagrama de funcionamiento de .htaccess

De esta sencilla forma, se asegura que adapte todas las páginas HTML del árbol de directorios, porque el fichero .htaccess debe ser situado en el directorio raíz desde el que cuelgan el resto de directorios que albergan las páginas HTML, para que afecte a todos ellos, sin tener que situar un fichero .htaccess en cada carpeta.

#### 3.4.5.6 Rendimiento de la adaptación automática de la página

En este punto la solución ofrecida para el problema planteado ha demostrado ser eficaz. Es por eso que tras desarrollar todo el sistema se decidió realizar unas pruebas del

mismo, con el objetivo de determinar el rendimiento. Estas pruebas consisten en la ejecución de un script PHP, el cual realiza un número determinado de veces la adaptación de la página HTML que se le indica. Al ejecutar este sencillo script se utiliza el comando de linux “time”, el cual muestra el tiempo que se ha consumido en realizar la tarea.

A continuación se puede ver el script utilizado:

```
#!/bin/bash
let count=$2
ua="Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)"
for ((i=0; i<count;i+=1)); do
    wget -q -O out.txt --user-agent="$ua" "$1"
done
```

El funcionamiento del script es muy sencillo. En el momento de la ejecución del comando se le pasan dos argumentos al script, el primero es la página que se quiere adaptar, \$1, y el segundo es el número de veces que se quiere realizar la operación. Cada vez que se quiera simular que se esta utilizando un dispositivo distinto, simplemente se tiene que modificar la variable \$ua del script. Así, el script realiza un bucle en el que obtiene la página pasada como parámetro, mediante la herramienta “wget”, la cual permite definir el user-agent, con lo que se consigue simular peticiones desde dispositivos móviles. De esta forma, el servidor recibe peticiones, las cuales entran en todo el sistema que se ha desarrollado en el proyecto, detección de dispositivo y adaptación de la página al mismo.

Las pruebas han consistido en simular las peticiones desde varios dispositivos, pertenecientes tanto a la categoría de retrato, como a la de paisaje. Igualmente, se ha probado con varios navegadores web de ordenador. Se han adaptado 3 páginas distintas para cada dispositivo, la página de inicio, la página de ayuda y la página de las tarifas roaming. Al probar con distintas páginas, se pretende comprobar como afecta la estructura propia de una página al rendimiento de la adaptación. Para todos los casos descritos, al ejecutar el script, se ha definido el parámetro del número de repeticiones en 1 y 100 veces, para una mayor fiabilidad de los resultados. Además, para cada dispositivo, la prueba en la que se ha realizado 1 sola vez la adaptación, era la primera vez que ese dispositivo solicitaba una página al servidor. Este hecho es importante, puesto que la base de datos WURFL, como se explicó en su momento, dispone de un sistema de caché, que almacena los dispositivos que ya han sido utilizados, para realizar consultas futuras más rápidamente. Por esto, es de esperar un mejor resultado, de media, en las ocasiones en que se han realizado 100 repeticiones que cuando se ha realizado la primera petición.

En la tabla 6 se muestran los resultados de todas las pruebas realizadas, medido el tiempo en segundos.

Categoría	Dispositivo	Página de inicio		Página de ayuda		Pág. tarifa roaming	
		1ª pet.	100 pet.	1ª pet.	100 pet.	1ª pet.	100 pet.
Retrato	iPhone	0,24	19,092	0,313	28,26	0,13	12,1
Retrato	MDA Vario	0,262	18,941	0,325	28,09	0,14	11,9
Retrato	MDA III	0,209	19,172	0,302	28,16	0,14	12
Retrato	Nokia e61	0,319	19,057	0,302	28,11	0,15	12
Retrato	Nokia n80	0,269	18,983	0,307	28,02	0,15	11,8
<b>Retrato</b>	<b>Media</b>	<b>0,2598</b>	<b>19,049</b>	<b>0,31</b>	<b>28,13</b>	<b>0,14</b>	<b>12</b>
Paisaje	Nokia n810	0,464	20,177	0,432	28,39	0,19	12,1
Paisaje	Toshiba g910	0,346	19,67	0,471	28,36	0,24	11,9
<b>Paisaje</b>	<b>Media</b>	<b>0,405</b>	<b>19,9235</b>	<b>0,452</b>	<b>28,38</b>	<b>0,22</b>	<b>12</b>
PC	IE 8	0,197	7,358	0,101	7,437	0,1	7,41
PC	Firefox 3.5	0,098	7,335	0,102	7,395	0,1	7,34
<b>PC</b>	<b>Media</b>	<b>0,1475</b>	<b>7,3465</b>	<b>0,102</b>	<b>7,416</b>	<b>0,1</b>	<b>7,37</b>

**Tabla 6: Rendimiento de la adaptación automática (versión en PHP)**

En primer lugar se puede observar que, para la categoría de PC, los tiempos obtenidos son siempre iguales, y es que en estos casos no hay adaptación, hay solo detección del dispositivo. Así, en tratar 100 peticiones se tarda 7.4 segundos aproximadamente, independientemente de la página solicitada.

Se puede apreciar que, en los casos de solicitar la página un dispositivo móvil, es decir, que no pertenezca a la categoría de PC, la página que menos tiempo ha requerido para ser adaptada es la de las tarifas roaming, con 12 segundos para 100 peticiones, y la que más tiempo ha necesitado es la de ayuda, con 28 segundos para 100 solicitudes, más del doble de tiempo. La página de inicio ha tardado entre 19 y 20 segundos para ser adaptada 100 veces. Esto da las claves de lo que hace que una página tarde más o menos en ser adaptada, y es que la página de tarifas roaming es una página muy sencilla, con apenas contenido, mientras que la página de ayuda es muy básica en cuanto a variedad de elementos, ya que todo el contenido es texto, pero tiene mucho texto. Por otra parte, la página de inicio es más compleja que las otras dos páginas, por tener más variedad de elementos, pero no tiene tanto texto como la página de ayuda. Por tanto, según las pruebas realizadas, la cantidad de texto, la longitud de la



página, es un factor determinante a la hora de adaptar la página. A mayor cantidad de texto, más tiempo se tarda en adaptar la página.

Otra conclusión que se puede sacar de la tabla de rendimiento es que, la primera vez que el dispositivo realiza la petición, el proceso completo de adaptación requiere más tiempo que el resto de las veces, llegando en algunos casos a durar el doble de tiempo, hecho que había sido advertido anteriormente. Por poner un ejemplo, la media del tiempo que necesitan los dispositivos de la categoría de paisaje para 100 peticiones de la página de inicio es de 19,9 segundos, con lo que de media se necesitarían 0,199 segundos por cada petición, mientras que para la primera petición de la misma página se han necesitado 0,405 segundos, algo más del doble de tiempo.

Se puede observar también que las adaptaciones, cuando el dispositivo pertenece a la categoría de paisaje, requieren más tiempo que cuando el dispositivo pertenece a la categoría de retrato. En todos los casos, ya sea para la primera petición o para el conjunto de 100 peticiones, de las 3 páginas, el dispositivo de la categoría de paisaje que menos tiempo necesita para que se adapte su página, requiere más tiempo que el dispositivo de la categoría de retrato cuya adaptación tarda más en realizarse, o en algún caso extremo, requiere el mismo tiempo. Este hecho no puede ser anecdótico, sino que tiene que deberse a que, en la categoría de retrato, la página que se obtiene como resultado es más básica que la obtenida para la categoría de paisaje, e incluso tiene más elementos eliminados, lo que hace que requiera un menor tratamiento por parte del servidor. Y a menor tratamiento, menos tiempo requerido, lógicamente.

Todos los datos extraídos a través de estas pruebas de rendimiento tienen que tratarse con cuidado, puesto que el equipo que actúa como servidor es el portátil proporcionado por la compañía al inicio del periodo de prácticas. Este portátil tenía varios años, y distaba mucho de las características técnicas de un servidor real. Por eso estas pruebas tienen interés para comparar los datos obtenidos, y extraer algunas conclusiones, pero no para extrapolar estos datos a la realidad.

Ya que tras el desarrollo de todo el sistema de detección de dispositivos y adaptación de la página, el tutor de la compañía reprodujo todo el sistema, pero utilizando la librería Java de la base de datos WURFL en un contenedor de servlets Tomcat, en lugar de la librería PHP

en un servidor Apache, y con el objetivo de comparar los resultados de rendimiento expuestos anteriormente, se procedió a realizar unas pruebas en el mismo equipo que se realizaron todas las pruebas anteriores. Estas pruebas no fueron tan exhaustivas, ya que no se probaron todos los dispositivos, solo algún representante de cada categoría, ni se adaptaron las 3 páginas, sino la página que menos tiempo había requerido para la versión en PHP, y la que más tiempo requirió. La tabla 7 muestra los resultados obtenidos.

Categoría	Dispositivo	Página de ayuda		Pág. tarifa roaming	
		1ª pet.	100 pet.	1ª pet.	100 pet.
Retrato	iPhone	0,05	3,43	0,06	2,04
Retrato	MDA III	0,06	3,5	0,03	1,66
<b>Retrato</b>	<b>Media</b>	<b>0,05</b>	<b>3,46</b>	<b>0,05</b>	<b>1,85</b>
Paisaje	Nokia n810	0,07	3,41	0,04	1,86
<b>Paisaje</b>	<b>Media</b>	<b>0,07</b>	<b>3,41</b>	<b>0,04</b>	<b>1,86</b>
PC	IE 8	0,05	1,58	0,01	1,16
<b>PC</b>	<b>Media</b>	<b>0,05</b>	<b>1,58</b>	<b>0,01</b>	<b>1,16</b>

Tabla 7: Rendimiento de la adaptación automática (versión en Java)

Como se puede observar en la tabla, en todos los casos probados, los tiempos han sido menores al utilizar la versión en Java. Para poder comparar mejor estos resultados con los obtenidos anteriormente para la versión desarrollada para el proyecto en PHP, se incluye a continuación, la tabla 8 con los datos de ambas versiones, solo con los casos probados para la versión en Java. En dicha tabla, los valores cuya fuente es de color negro son los obtenidos con la versión en PHP, y los datos cuya fuente es de color azul, los obtenidos con la versión en Java:

Categoría	Dispositivo	Página de ayuda				Pág. tarifa roaming			
		1ª pet.		100 pet.		1ª pet.		100 pet.	
Retrato	iPhone	0,313	0,05	28,26	3,43	0,13	0,06	12,1	2,04
Retrato	MDA III	0,302	0,06	28,16	3,5	0,14	0,03	12	1,66
<b>Retrato</b>	<b>Media</b>	<b>0,31</b>	<b>0,05</b>	<b>28,13</b>	<b>3,46</b>	<b>0,14</b>	<b>0,05</b>	<b>12</b>	<b>1,85</b>
Paisaje	Nokia n810	0,432	0,07	28,39	3,41	0,19	0,04	12,1	1,86
<b>Paisaje</b>	<b>Media</b>	<b>0,452</b>	<b>0,07</b>	<b>28,38</b>	<b>3,41</b>	<b>0,22</b>	<b>0,04</b>	<b>12</b>	<b>1,86</b>
PC	IE 8	0,101	0,05	7,437	1,58	0,1	0,01	7,41	1,16
<b>PC</b>	<b>Media</b>	<b>0,1</b>	<b>0,05</b>	<b>7,414</b>	<b>1,58</b>	<b>0,1</b>	<b>0,01</b>	<b>7,37</b>	<b>1,16</b>

Tabla 8: Comparativa del rendimiento de las versiones PHP y Java de la adaptación automática

Al comparar los datos de la tabla, se puede observar que, para el caso en el que se realice la petición desde un PC, la versión en Java requiere menos tiempo para realizar la adaptación, para 100 peticiones, entre 1 segundo y 1 segundo y medio, mientras que la versión en PHP necesita casi 7 segundos y medio.

Para los casos en los que la solicitud viene desde un dispositivo móvil, probando la versión en Java, se puede apreciar que, al igual que pasaba con la versión en PHP, la página que menos tiempo requiere para ser adaptada es la de las tarifas roaming, con 1,8 segundos para 100 peticiones, mientras que la página que más tiempo necesita es la de ayuda, con 3,4 segundos para realizar 100 adaptaciones, con lo que las conclusiones extraídas con la versión en PHP se reafirman. No obstante, se puede observar que el rendimiento es mejor con la versión en Java, es decir, que los tiempos requeridos son menores. En el caso de la página de ayuda, para 100 peticiones, tarda aproximadamente 8 veces menos (28 segundos con la versión en PHP y 3,4 con la versión en Java), mientras que en el caso de la página de tarifas roaming, para 100 peticiones, se obtienen unos resultados casi 7 veces mejores (12 segundos con la versión en PHP y 1,8 con la versión en Java).

Otra conclusión que se reafirma, a la vista de los resultados de la versión en Java, es que la primera vez que el dispositivo hace la petición, la adaptación lleva más tiempo que el resto de las veces, llegando a ser más del doble en algunos casos determinados. Por ejemplo, la media del tiempo requerido para 100 peticiones de la página de tarifas roaming desde un dispositivo de la categoría de retrato es de 1,85 segundos, con lo que de media se necesitarían 0,018 segundos para una petición, mientras que para la primera petición el resultado ha sido de 0,05 segundos, casi el triple de tiempo.

Con la versión de Java no se puede concluir que las adaptaciones requieran más tiempo cuando el dispositivo pertenece a la categoría de paisaje que cuando pertenece a la categoría de retrato, ya que los datos obtenidos son muy similares en ambas categorías, siendo en algunos casos ligeramente superiores los de una categoría, y en otros casos los de la otra categoría. Este hecho podría deberse a que el mayor procesamiento que requiere la adaptación para la categoría de paisaje es inapreciable, debido a que la versión en Java es considerablemente más rápida que la versión en PHP. Con ésta, en todos los casos se requería más tiempo para adaptar la categoría de paisaje, pero para la primera petición se obtenía un

resultado más claro que para 100 peticiones, caso en el que la categoría de paisaje requería poquísimos tiempo más.

## Capítulo 4 - Conclusiones y trabajos futuros

En este apartado se analizan las conclusiones que se pueden extraer tras la realización del proyecto, y se exponen los siguientes pasos que podrían tomarse para mejorar el actual sistema que se ha desarrollado, así como las tareas en las que se podría trabajar en el futuro para extender su funcionalidad.

### 4.1 Conclusiones

Volviendo la vista atrás para recordar los objetivos marcados al principio del proyecto, se puede analizar si estos se han cumplido o no. En primer lugar se analizó el problema que provocó la necesidad de realizar este proyecto, así como las posibles soluciones que existían en la actualidad. Tras esta puesta en situación se pudieron evaluar las distintas soluciones que se propusieron para centrarse finalmente en la que mas satisfizo los requisitos. En esta primera parte, no tendría cabida solo lo tratado en el capítulo Estado del Arte respecto a los transcodificadores, sino también todos los estudios realizados referentes a las características de los dispositivos, el análisis de la página con la que se tiene que trabajar, etc.

La segunda parte del proyecto es la referente al desarrollo de la solución planteada, para obtener un prototipo ajustado al máximo al entorno del sistema real. De esta forma, se ha desarrollado un prototipo que puede ser mostrado en presentaciones, el cual es totalmente independiente, y no requiere de nada, puesto que incluye hasta una copia de la página original del hotspot, estando preparado para que cualquier dispositivo se conecte a él, y poder apreciar los resultados.

En cuanto a la estructura del proyecto, éste se ha desarrollado de forma modular para que las distintas partes en las que está dividido pudieran utilizarse por separado. De esta forma, se podría utilizar el módulo de la detección de dispositivos, o el de la adaptación de la página, de forma independiente, si en un futuro la compañía así lo requiere. Desde el principio del proyecto, el tutor en la compañía sugirió realizarlo así, teniendo en mente el poder utilizar en el futuro el proyecto completo o uno de los módulos por separado, debido a que no se conocía exactamente el entorno real en el que se tendría que instalar el sistema desarrollado

en este proyecto. De hecho, durante la realización del proyecto se barajó la posibilidad de que, desde el departamento que mantiene el hotspot, solo estuvieran interesados en la parte de la detección del dispositivo, o que ya tuvieran una forma de detectar las características de los dispositivos, y solo requirieran la parte de la adaptación de la página.

Por otra parte, se han definido unas recomendaciones para el editor de la página de entrada al punto de acceso, para que, con unos añadidos a la hora de crear la página original, se pueda predecir el comportamiento del sistema de adaptación automática para los dispositivos móviles, es decir, se pueda obtener las páginas adaptadas con solo tener cuidado al crear el diseño de la página original.

No obstante, se ha respetado el requisito original de no modificar directamente la página web original. Y es que, podría pensarse que lo más sencillo es diseñar nuevas páginas para los dispositivos, a partir de la original, pero la idea general del proyecto es utilizar la página original para obtener automáticamente una versión móvil de la misma, lo cual es distinto.

Durante la realización del proyecto, en el departamento de investigación y desarrollo de Deutsche Telekom en Berlín, se ha mantenido contacto con el departamento que mantiene los hotspots, con lo que se ha podido tener más datos referentes al entorno real de los puntos de acceso, hecho importante, ya que al principio del proyecto se desconocía el sistema real que, desde dicho departamento, se utilizaba para mantener los puntos de acceso. En algún momento del desarrollo del proyecto se sacaron conclusiones erróneas acerca de la forma del sistema real que se usaba para mantener los hotspots, y es que el departamento que lo mantiene se hallaba situado en otra ciudad distinta, con lo que la comunicación no podía ser tan fluida como hubiera sido deseado, pero finalmente, como fruto de los contactos mantenidos se pudo conocer como funcionaba todo el sistema real.

A pesar de estos inconvenientes, se pudo mantener contacto con gente implicada en el desarrollo de los portales de los hotspots, hasta se prepararon un par de presentaciones con ellos para conocer sus impresiones acerca del proyecto desarrollado, así como para obtener nuevos requisitos por su parte, los cuales hubo que satisfacer para que la solución planteada se amoldara al entorno real. Incluso se pudo participar en una reunión interna del departamento, en calidad de oyente, en la que se mostraba el funcionamiento de nuevas herramientas para la

creación de contenidos, tanto para la página de inicio del hotspot, como para el portal de Internet.

Por otra parte, a nivel personal, se ha realizado el proyecto en una compañía puntera de un país a la última, tecnológicamente hablando, como es Alemania, con lo que se ha podido conocer la forma de trabajar en un entorno real e internacional, lo cual es, sin lugar a dudas, muy enriquecedor.

## **4.2 Trabajos futuros**

Un aspecto muy interesante, y que no se pudo tener en cuenta cuando se realizó el proyecto, ya que no se encontró la forma de hacerlo, es considerar el tamaño de la pantalla. El tamaño de la pantalla se mide habitualmente en pulgadas, y es distinto de la resolución de la pantalla, la cual se mide en píxeles. Esta característica es muy importante a la hora de visualizar algo en la pantalla de un móvil, y a medida que pase el tiempo lo será más, ya que los dispositivos móviles tienen cada vez más resolución, tanta como la tecnología actual se lo permita, y esto es siempre una cualidad. Pero, por otro lado, el tamaño de las pantallas tiene un límite, puesto que, por definición, un dispositivo móvil ha de tener un tamaño que le permita seguir siendo “móvil”. Se puede observar un ejemplo con el nuevo iPhone 4, el cual tiene una resolución de pantalla de 640 x 960, mientras que la de su antecesor, el iPhone 3GS, es de 320 x 480, es decir, la mitad. El dato relevante es que ambos modelos tienen el mismo tamaño de pantalla. Con este ejemplo se puede observar que, sin tener en cuenta las posibles modificaciones que realicen los dispositivos, una página creada con una determinada resolución ocupará la mitad de espacio en la pantalla del nuevo modelo. De esta forma, se puede ver que es necesario controlar el tamaño de la pantalla de los dispositivos, junto a la resolución, para ofrecer una adaptación de la página más precisa. Durante la realización del proyecto se advirtió de este aspecto, pero no se pudo encontrar una forma de obtener el tamaño de la pantalla del dispositivo, y en la base de datos de dispositivos WURFL no aparece esta característica. Es de desear, que en el futuro haya alguna manera fiable de averiguar el tamaño de la pantalla.

Otro aspecto importante que se debería tener en cuenta para los trabajos futuros es el que se refiere al acelerómetro que incorporan algunos móviles. El acelerómetro de un móvil

permite, entre otras cosas, que éste conozca la orientación en la que se encuentra el dispositivo. Así, si se gira el dispositivo, para cambiar de la orientación vertical normal a la orientación “apaisada”, lo que esté apareciendo en pantalla deberá girar también. El gran inconveniente que plantea esto, en lo que respecta al proyecto desarrollado, es que, en el momento del giro, la anchura pasa a ser la altura y viceversa, pero, en el estado final del proyecto, la página que se estaba visualizando había sido adaptada para la anchura y altura original, con lo que, al realizar el giro, la página seguirá teniendo las mismas dimensiones que antes de realizar el giro y, obviamente, no estará ajustada a la nueva anchura y altura. En la base de datos de dispositivos WURFL aparece una característica que permite conocer si el dispositivo es capaz o no de mostrar la pantalla en ambas orientaciones, pero esto no sirve para saber cuál es la orientación actual del móvil, hecho que debería averiguarse de otro modo. No obstante, si se consiguiera incluir esta funcionalidad en el proyecto desarrollado, se plantearía una nueva problemática, y es que quizá el rendimiento no sea óptimo al realizar todo el proceso de detección del dispositivo y adaptación de la página cada vez que el usuario voltee el terminal, es decir, volver a realizar todo el proceso a pesar de que el usuario permanezca en la misma página web, en lugar de hacerlo solamente cuando el usuario cambie de página web.

En la última reunión mantenida con gente del departamento responsable del hotspot, se conoció el uso de una tecnología llamada WISPr (Wireless Internet Service Provider roaming). Según esta gente, gracias a esta tecnología, no debería ser necesario adaptar las páginas web una y otra vez para un mismo dispositivo, ya que éste permanece conectado a los puntos de acceso Wi-Fi sin tener que autenticarse manualmente en ellos. En esos casos, se recibe un user-agent específico de WISPr, con lo que se podría controlar estos casos. No obstante, debido a que el periodo de prácticas terminó días después de esta última reunión, no se pudo investigar nada acerca de esta tecnología WISPr, pero sería interesante hacerlo en el futuro, ya que ésta fue una recomendación directa del departamento responsable del hotspot, para mejorar el funcionamiento o el rendimiento actual del proyecto.

En la primera versión del script de adaptación, la forma de reorganizar los elementos de la página web era directamente mediante llamadas a funciones, que indicaban de forma explícita, qué elemento se situaba a continuación de qué elemento. Posteriormente, en la versión final, esto se modificó para que, en el momento del diseño de la página original se incluyeran prioridades de los elementos. Éstas consistían en un número empezando desde el



número 1. De esta forma, en el script de adaptación, simplemente se dice, mediante las mismas funciones de antes, que el elemento con prioridad “x” va a continuación del elemento con prioridad “y”, es decir, el de prioridad uno delante del de prioridad dos, y así sucesivamente. Esta modificación hace que no sea necesario el conocimiento de la semántica de la página web original, ya que solo se trata con prioridades. No obstante, este sistema tiene una limitación, se ha de llamar una vez más a la función si se incluye un nuevo elemento en la página original. Además, dado que las prioridades definirán el orden de los elementos, el trabajo futuro que podría mejorar el sistema actual es incluir un método de ordenación que realizara lo mismo automáticamente. Además se elimina la limitación de tener que incluir la llamada a la función en el script de adaptación, en caso de añadir nuevos elementos a la página web original.

Como trabajo futuro se podría afinar la forma de clasificar los dispositivos en las categorías. Dotando de una mayor inteligencia a la condición que define a qué categoría pertenece un dispositivo, se obtendrían unos resultados de adaptación más precisos. Además, realizar esta modificación es muy sencillo, no hay más que agregar las nuevas condiciones al script de detección del dispositivo. Por ejemplo, se podría valorar la conveniencia de incluir una condición que clasificara a los dispositivos móviles con una resolución lo suficientemente alta en la categoría de ordenadores personales, con lo que se vería la página original sin realizar ninguna adaptación, siempre y cuando la visualización de ésta en dichos dispositivos fuera la óptima.

Otro aspecto que se podría estudiar en el futuro es el referente a la creación de nuevas categorías. Actualmente, aparte de la categoría de PC, los dispositivos móviles se clasifican en las categorías de retrato y de paisaje. Se podría analizar si es conveniente definir alguna categoría más, o añadir alguna sub-categoría dentro de las categorías actuales, para un mejor resultado.

En cuanto a las posibilidades de implantación del sistema desarrollado en el proyecto, se podría estudiar la posibilidad de utilizar esta forma de adaptar la página de acceso al hotspot en otros servicios que los usuarios de dispositivos móviles utilicen, como pueden ser el correo electrónico de Deutsche Telekom, el portal de Internet, noticias, o incluso la propia intranet de la compañía, para así ofrecer un servicio de valor añadido que hará al usuario

sentirse valorado por la compañía, al ofrecérsele todos estos servicios adaptados y ajustados a su dispositivo móvil.

## Apéndice A - Presupuesto

### A.1 *Descomposición del proyecto en tareas*

El trabajo realizado en este proyecto se puede dividir en tareas, las cuales fueron definidas en el apartado Objetivos de la presente memoria. Además se han incluido sub-tareas en el caso de las tareas originales más importante, para una mejor clasificación del tiempo requerido. A continuación se puede ver un listado de las tareas en las que se ha dividido todo el trabajo:

- **Análisis del problema:** Consiste en hacer pruebas del sistema actual con distintos dispositivos y analizar los resultados para definir el problema. Tiene una duración de 1 semana.
- **Estado del arte:** Consiste básicamente en analizar profundamente las distintas posibilidades existentes en el mercado para solucionar el problema planteado de la adaptación de sitios web para usuarios móviles, y discutir la conveniencia de utilizar alguna de ellas. El tiempo empleado en esta tarea ha sido de 1 mes.
- **Desarrollo solución:** Consiste en desarrollar la solución planteada a modo de prototipo. Tiene una duración de 14 semanas. Se compone de las siguientes sub-tareas:
- **Detección del dispositivo:** Incluye todo el estudio referente a las distintas posibilidades de detectar las características de los dispositivos, así como la explicación detallada en todo lo que respecta a la base de datos WURFL. Esta tarea tiene una duración de 3 semanas.
- **Definición categorías y estudios previos:** Consiste en los estudios realizados para la definición de las categorías, por ejemplo, características más relevantes de los dispositivos, tendencia de las resoluciones de pantallas de los dispositivos, etc. Incluye la propia definición de las categorías. Tiene una duración de 2 semanas.
- **Análisis página de inicio:** Consiste en analizar la página de inicio para conocer los elementos que la componen, y poder concluir cuáles son más importantes. También se analizan las tecnologías de las que hace uso la página. Tiene una duración de 1 semana.

- Rediseño estático: Consiste en rediseñar manualmente la página web para cada categoría, para obtener un boceto de cómo debería ser el resultado final de la adaptación. Tiene una duración de 2 semanas.
- Adaptación automática: Consiste en realizar un script que permita obtener la página adaptada para cada categoría, teniendo como entrada la página original. Es la parte que requiere más tiempo, 6 semanas.
- Pruebas: Incluye todas las pruebas realizadas con dispositivos durante la realización del proyecto, por ejemplo, pruebas del sistema de detección de dispositivos, pruebas del rediseño estático de la web, pruebas finales, etc. Tiene una duración estimada de 2 semanas.
- Presentaciones: Consiste en todas las presentaciones y reuniones llevadas a cabo durante la realización del proyecto, así como la preparación de las mismas. Tiene una duración aproximada de 1 semana.
- Redacción memoria: Consta del trabajo de documentación de todo el trabajo realizado en el proyecto. Tiene una duración de 6 semanas, de las cuales 2 de ellas han sido dentro de los 6 meses del periodo de prácticas, y las otras 4 al término del mismo.

## **A.2 Costes del proyecto**

Como datos generales, se han considerado jornadas laborales de 7,5 horas al día y 5 días a la semana. El sueldo de ingeniero considerado a la hora de hacer los cálculos ha sido de 30 euros por hora de trabajo.

Se incluye el concepto viajes, que aparece reflejado en la tabla siguiente. Este concepto se refiere a los viajes realizados para mantener reuniones con el departamento encargado de los puntos de acceso Wi-Fi.

Igualmente, en el presupuesto, se incluye el material que ha sido necesario para llevar a cabo el proyecto. En este caso, un portátil en el que desarrollar el proyecto, y dispositivos móviles con los que poder realizar las pruebas.

A continuación, en la tabla 9, se incluye un desglose de todos los aspectos referentes al presupuesto del proyecto.

Concepto	Cantidad	Precio	Importe
<b>Análisis del problema</b>	37.5 horas	30 € / hora	1125 €
<b>Estado del arte</b>	150 horas	30 € / hora	4500 €
<b>Desarrollo solución</b>			
-Detección del dispositivo	112.5 horas	30 € / hora	3375 €
-Definición categorías y estudios previos	75 horas	30 € / hora	2250 €
-Análisis página de inicio	37.5 horas	30 € / hora	1125 €
-Rediseño estático	75 horas	30 € / hora	2250 €
-Adaptación automática	225 horas	30 € / hora	6750 €
<b>Pruebas</b>	75 horas	30 € / hora	2250 €
<b>Presentaciones</b>	37.5 horas	30 € / hora	1125 €
<b>Redacción memoria</b>	225 horas	30 € / hora	6750 €
<b>Viajes</b>	2 viajes	600 €	1200 €
<b>Material</b>			
-Portátil	1 unidad	500 €	500 €
-Dispositivos para pruebas	Varios	Variado	400 €
<b>Total</b>	33600 €		

**Tabla 9: Presupuesto final del proyecto**

Como se puede observar en la tabla, el presupuesto total del proyecto asciende a un total de 33600 euros.

## Apéndice B – Resultados de la adaptación

A continuación se incluyen fotos tomadas a varios de los dispositivos móviles que se han utilizado para realizar las pruebas. Para todos los casos se incluye la imagen del terminal con la página web original, junto a otra imagen del mismo terminal con la página resultante de todo el proceso automático de adaptación que se describe en la presente memoria, para poder apreciar las diferencias en un entorno real.

### B.1 iPhone:

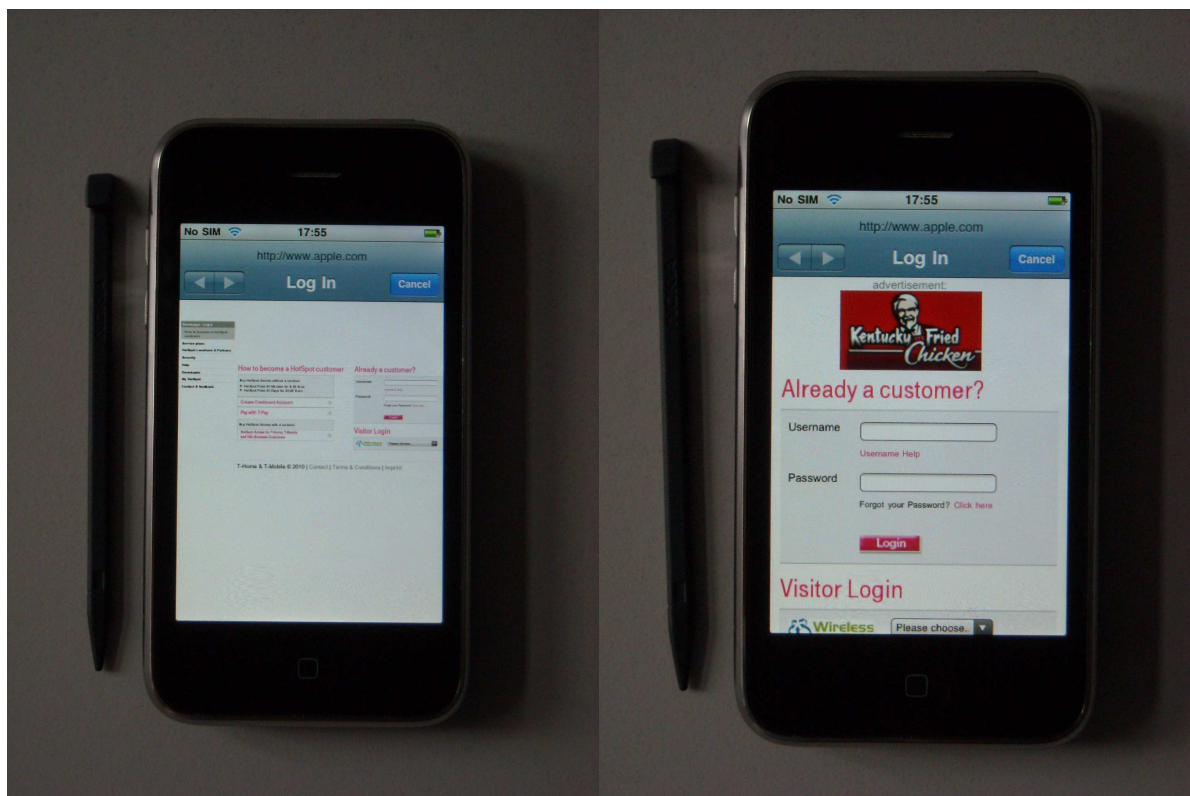


Figura 24: Página original (izda) y página resultante de la adaptación (dcha) en un iPhone

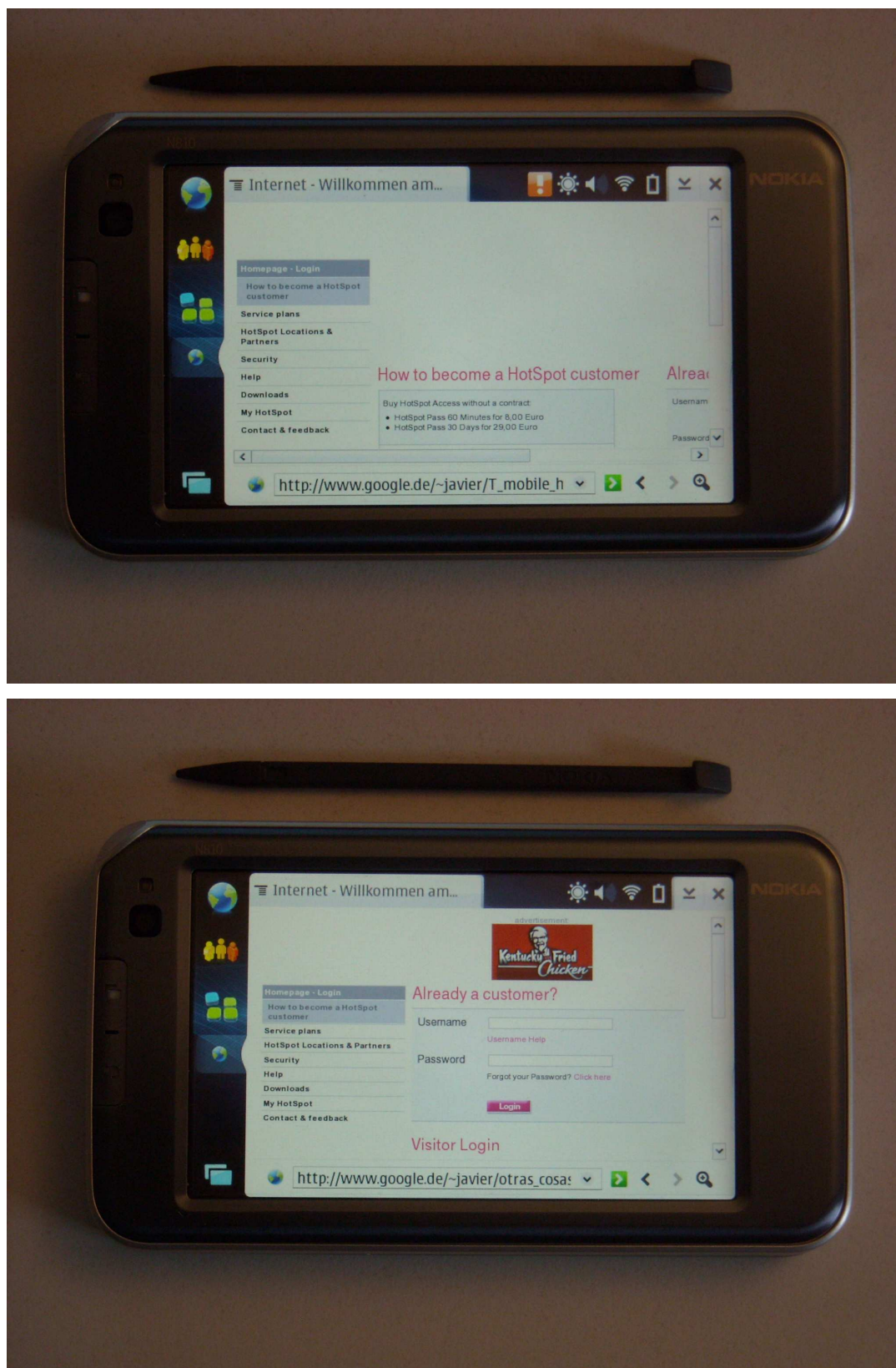
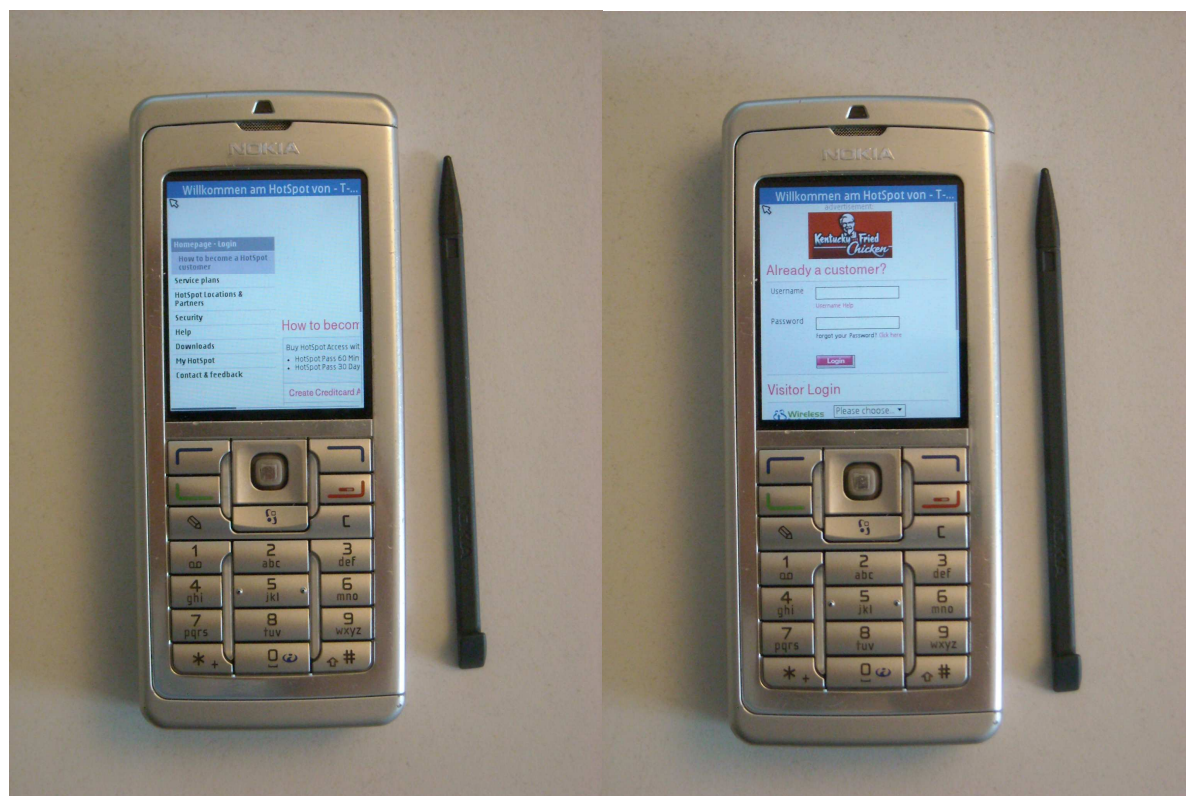
**B.2 Nokia N810:**

Figura 25: Página original (arriba) y página resultante de la adaptación (abajo) en un Nokia N810



**B.3 Nokia E60:**

**Figura 26:** Página original (izda) y página resultante de la adaptación (dcha) en un Nokia E60



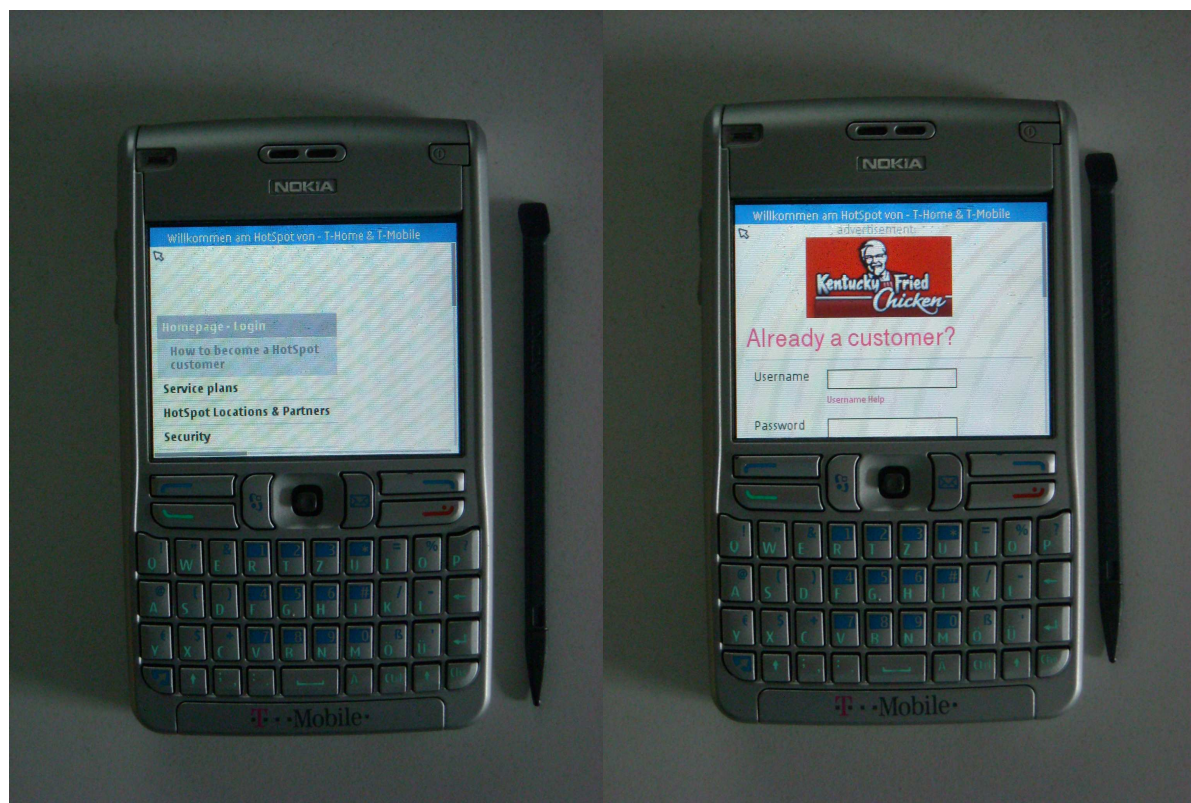
**B.4 Nokia N61:**

Figura 27: Página original (izda) y página resultante de la adaptación (dcha) en un Nokia N61

## Apéndice C – Código de los script de adaptación y funcionamiento de todo el proceso

En este apéndice se incluye todo el código PHP desarrollado para realizar la adaptación automática de la página, junto a las explicaciones necesarias, tanto de las funciones como del proceso que se va llevando a cabo.

### C.1 *Script para la detección del dispositivo*

En primer lugar, cuando un terminal intenta acceder a una de las páginas, el fichero .htaccess que ya ha sido explicado en la memoria, guarda la dirección de la página que el usuario desea visitar y reescribe la URL, para conducirle al script que realiza la detección del dispositivo, cuyo código PHP es el siguiente:

```
<?php
# include the script where the variable $wurflmanager is defined
include_once '/home/javier/public_html/wurfl-php-1.r1/examples/demo2/inc/header.inc';
# include adaptation_portrait and adaptation_landscape functions
include_once '/home/javier/public_html/otras_cosas/T_mobile_hotspot_dinamico/hotspot.t-
mobile.net/wlan/adaptation_tool_portrait.php';
include_once '/home/javier/public_html/otras_cosas/T_mobile_hotspot_dinamico/hotspot.t-
mobile.net/wlan/adaptation_tool_landscape.php';
# include DOM API to treat HTML as a logical tree
include('/home/javier/public_html/otras_cosas/T_mobile_hotspot_dinamico/hotspot.t-
mobile.net/wlan/dom/simple_html_dom.php');
# include functions to operate the DOM tree in an easier way
include("/home/javier/public_html/otras_cosas/T_mobile_hotspot_dinamico/hotspot.t-
mobile.net/wlan/functions.php");

ob_start(); //start buffer capture

# get the device and its characteristics
$requestingDevice = $wurflManager->getDeviceForHttpRequest($_SERVER);

$resolution_width = $requestingDevice->getCapability("resolution_width");
$is_wireless = $requestingDevice->getCapability("is_wireless_device");
$claims_web_support = $requestingDevice->getCapability("device_claims_web_support");
if (($is_wireless == "false") && ($claims_web_support == "true"))
{
    # the request is from a desktop PC
    header("Location: /~javier/T_mobile_hotspot/hotspot.t-
mobile.net/wlan/index.do%3flang=en_GB.html");
} else
{
    # the request is from a mobile device
```

```

$file = $_GET["file"];
if (!$file)
    $file = "~/javier/otras cosas/T_mobile_hotspot_dinamico/hotspot.t-
mobile.net/wlan/index.do?lang=en_GB.html";
$resolution_width_int = (int) $resolution_width;
if ($resolution_width_int < 500)
{
    # the request is from a mobile device, portrait category
    adaptation_portrait( $resolution_width, $file );

} else
{
    # the request is from a mobile device, landscape category
    adaptation_landscape( $resolution_width, $file );
}
}
ob_end_flush(); //end buffer capture
?>

```

Como se puede observar en el código, en caso de que la solicitud venga desde un dispositivo móvil, éste se clasifica en su categoría correspondiente y se llama a la función de adaptación adecuada, pasando como parámetros la resolución del terminal y la página concreta que se ha solicitado.

## ***C.2 Función que adapta la página para la categoría de retrato***

A continuación se incluye la función `adaptation_portrait` que, haciendo uso de determinadas funciones, adapta la página solicitada para el dispositivo que realiza la petición, y la muestra por pantalla, sin hacer redirecciones:

```

<?php
function adaptation_portrait( $width, $file ) {

    $category = "portrait";

    # set the minimum width. For smaller screens the width will be 200px to keep the adapted
    page clear
    $minimum_width = 200;

    # get the width of the device passed
    $resolution_width = $width;

    # get DOM from URL passed
    $file = str_replace( "~/javier", "home/javier/public_html", $file );
    $html = file_get_html($file);

    # perform the changes for portrait category
    add_iphone_meta_tag($html);
    change_css($html, $category);
    hide_elements($html, $category);
}

```

```

reorder_elements($html, "", 4, 3);
reorder_elements($html, "", 3, 2);
reorder_elements($html, "", 2, 1);

# set dinamycally the global width of the page, according to the device which is accessing to
the page
if (($resolution_width * 0.9) > ($minimum_width)){
    modify_element($html, "layer_middle", $resolution_width * 0.95, 0, 0);
} else {
    modify_element($html, "layer_middle", $minimum_width, 0, 0);
}

$string = $html -> save();
echo $string;
}
?>

```

### ***C.3 Función que adapta la página para la categoría de paisaje***

La función de adaptación para la categoría de paisaje es muy similar, como se puede ver a continuación:

```

<?php
function adaptation_landscape( $width, $file ) {

    $category = "landscape";

    # get the width of the device passed
    $resolution_width = $width;

    # get DOM from URL or file
    $file = str_replace( "~javier", "home/javier/public_html", $file );
    $html = file_get_html($file);

    change_css($html, $category);
    hide_elements($html, $category);
    reorder_elements($html, "content_", 2, 1);
    reorder_elements($html, "", 4, 2);
    reorder_elements($html, "", 2, 1);

    # set dinamycally the global width of the page, according to the device which is accessing to
the page
    modify_element($html, "layer_middle", $resolution_width * 0.8, 0, 0);

    $string = $html -> save();
    echo $string;
}
?>

```

### ***C.4 Funciones auxiliares para realizar modificaciones en la página***

Ambas funciones de adaptación son muy sencillas, pues basan su funcionamiento en llamadas a funciones que son las que realizan las modificaciones en la página, utilizando funciones de más bajo nivel del árbol DOM. Estas funciones, que han sido desarrolladas durante la realización del proyecto, están incluidas en otro fichero, junto a otras que añaden funcionalidad que no se usan en la versión final del proyecto, pero que fueron utilizadas anteriormente. Dicho fichero se incluye a continuación:

```
<?php
# Add the meta tag "viewport" to make the web fits better to the iPhone display
function add_iphone_meta_tag($html){
    $meta_element = $html->find("meta", 0);
    $meta_element -> outertext = "<meta name = \"viewport\" content = \"width=device-width;
initial-scale=1.02; maximum-scale=1.02; user-scalable=0;\" />" . $meta_element -> outertext;
}

# Change the CSS file
function change_css($html, $css){
    foreach($html->find("link") as $element){
        if ($element -> outertext == "<link href=\"../TCOM/en_GB/web/css/styles.css\"
rel=\"stylesheet\" type=\"text/css\" />") {
            $element -> outertext = "<link href=\"../TCOM/en_GB/web/css/styles_\" . $css .
\".css\" rel=\"stylesheet\" type=\"text/css\" />";
        }
        if ($element -> outertext == "<link href=\"../css/styles.css\" rel=\"stylesheet\"
type=\"text/css\" />") {
            $element -> outertext = "<link href=\"../css/styles_\" . $css . ".css\"
rel=\"stylesheet\" type=\"text/css\" />";
        }
    }
}

# Hide all the elements (divs or list items) with the attribute $category_hide=yes
function hide_elements($html, $category){
    $elements = $html -> find("div [\".$category.\"_hide=yes],li [\".$category.\"_hide=yes");
    foreach ($elements as $element){
        $element -> outertext = " ";
    }
}

# Paste the element with lower priority on the element with higher priority. $parent_element can be
"content_" or "".
function reorder_elements($html, $parent_element, $lower_priority_element,
$higher_priority_element) {
    $element_lower = $html -> find("div [\".$parent_element.\"priority=\".$lower_priority_element.\""],
0);
    $element_higher = $html -> find("div
[\".$parent_element.\"priority=\".$higher_priority_element.\""], 0);
    $element_outertext = $element_lower -> outertext;
    $element_lower -> outertext = " ";
    $element_higher -> outertext = $element_higher -> outertext . $element_outertext;
}

# modify the design of an element by injecting a "style=..."
function modify_element($html, $element_name, $width, $height, $min_height){
    $element = $html -> find("div [id=\"\".$element_name.\"\""], 0);
    $string = $element -> outertext;
```

```

$string_modified = "";
$modified = 0;
$i = 0;
while ($i < strlen($string)) {
    $character = $string[$i];
    if (($character == ">") && (!$modified)){
        if ($height > 0)
            $string_modified .= " style=\"width: " . $width . "px;\" . \" height:
\".$height.\"px;\" . \" min_height: " . $min_height . "px;\"";
        else $string_modified .= " style=\"width: " . $width . "px;\" . \" min-height: " .
$min_height . "px;\"";
        $modified = 1;
    }
    $string_modified .= $character;
    $i++;
}
$element -> outertext = $string_modified;
}

# save an element in a file in the folder "elements"
/*This function shouldn't be used anymore because several clients accessing at the same time could
be modifying the same files they are using, so these kind of transactions should be done on the fly */
function save_element($element, $file) {
    $adap_file_handle = fopen("./elements/".$file.".php", "w");
    $result = fwrite($adap_file_handle, $element);
    fclose($adap_file_handle);
    return $result;
}

# write the "include" sentence of an element
function encapsulate_element($element, $file){
    return $element -> outertext = "<?php include(\"elements/".$file.".php\"); ?>";
}

# save the element in a file and change its code in the html for an include
function treat_element($html, $element_name){
    $element = $html -> find("div [id=\"".$element_name."\"]", 0);
    save_element($element, $element_name);
    encapsulate_element($element, $element_name);
}

# Delete an element
function delete_element($html, $element_name){
    $element = $html -> find("div [id=\"".$element_name."\"]", 0);
    $string = $element -> outertext;
    $element -> outertext = " ";
    return $string;
}

# Place an element as the next sibling of other element
function rearrange_element($html, $element_name, $previous_element_name) {
    $element_to_rearrange = delete_element($html, $element_name);
    $element = $html -> find("div [id=\"".$previous_element_name . "\"]", 0);
    $element -> outertext = $element -> outertext . $element_to_rearrange;
}
?>

```

## Apéndice D – Script rendimiento WURFL

A continuación se incluye el script PHP que se utilizó para medir el rendimiento de la base de datos WURFL, como ya se ha explicado anteriormente en la memoria:

```
<?php
include_once './inc/header.inc';

$cont = 100;

$time_start = microtime(true);
for ($size=1;$size<=$cont;$size++)
{
    $ua = $_SERVER["HTTP_USER_AGENT"];
    $requestingDevice = $wurflManager->getDeviceForUserAgent($ua);
}
$time_end = microtime(true);

$time1 = $time_end - $time_start;

$time_start = microtime(true);
for ($size=1;$size<=$cont;$size++)
{
    $ua = $_SERVER["HTTP_USER_AGENT"];
    $requestingDevice = $wurflManager->getDeviceForUserAgent($ua);
}
$time_end = microtime(true);

$time2 = $time_end - $time_start;

$time_start = microtime(true);
for ($size=1;$size<=$cont;$size++)
{
    $ua = $_SERVER["HTTP_USER_AGENT"];
    $requestingDevice = $wurflManager->getDeviceForUserAgent($ua);
}
$time_end = microtime(true);

$time3 = $time_end - $time_start;

echo "It took $time1, $time2 and $time3 seconds <br><br>";
$average = ($time1 + $time2 + $time3)/3;
echo "Average time = $average for $cont requests";
?>
```

## Referencias

- [1] Página del editor de imágenes GIMP, [www.gimp.org.es](http://www.gimp.org.es)
- [2] Página del Simple HTML DOM, <http://simplehtmldom.sourceforge.net/>
- [3] Página del World Wide Consortium, <http://www.w3c.org>
- [4] Elliotte Rusty Harold, “XML : Extensible Markup Language”. Foster City, IDG Books Worldwide. 1998
- [5] Página de las hojas estilos CSS, <http://www.w3.org/Style/css>
- [6] Página del servidor Apache, <http://httpd.apache.org>
- [7] Rich Bowen, Ken Coar., “Servidor Apache Al Descubierto”. Prentice Hall, 2000
- [8] Página principal del GNU, <http://www.gnu.org/>
- [9] Página de descarga de WURFL, <http://sourceforge.net/projects/wurfl/files/WURFL-PHP/>
- [10] Página del proyecto WURFL, <http://wurfl.sourceforge.net/>
- [11] Página del editor HTML KompoZer, <http://kompozer.net/>
- [12] Página del estudio acerca de la tendencia de la resolución de pantalla de los móviles, <http://sender11.typepad.com/sender11/2008/04/mobile-screen-s.html>
- [13] Página transcodificador basado en RSS, [www.mob-it.com](http://www.mob-it.com)
- [14] Página transcodificador basado en RSS, [www.feedm8.com](http://www.feedm8.com)
- [15] Página del navegador móvil Opera mini, <http://www.opera.com/mobile/>
- [16] Página del navegador móvil Teashark, <http://www.teashark.com/>
- [17] Página del transcodificador Google Mobilizer, [www.google.com/gwt/n](http://www.google.com/gwt/n)
- [18] Página del transcodificador Mowser, <http://mowser.com>
- [19] Página del transcodificador Skweezer, <http://www.skweezer.com/>
- [20] Página de la herramienta wget, <http://www.gnu.org/software/wget/>
- [21] Página de la base de datos de dispositivos DeviceAtlas, <http://deviceatlas.com/>
- [22] Página de la base de datos de dispositivos DetectRight, <http://www.detectright.com/>
- [23] Página de descarga del complemento “user-agent switcher” para Firefox, <https://addons.mozilla.org/es-ES/firefox/addon/59/>
- [24] .mobi, “Mobile Web Developer’s Guide - Part I: Creating simple mobile sites for common handsets”, 2007
- [25] Página del emulador de móviles Nokia, [http://www.forum.nokia.com/Devices/Remote\\_device\\_access/](http://www.forum.nokia.com/Devices/Remote_device_access/)



- [26] Gail Rahn Frederick with Rajesh Lal, “Beginning Smartphone Web Development: Building JavaScript, CSS, HTML, and Ajax-Based Applications for iPhone, Android, Palm Pre, Blackberry, Windows Mobile, and Nokia S60”, Apress 2009